ALGORITHMS FOR EFFICIENT UTILIZATION OF WIRELESS BANDWIDTH AND TO PROVIDE QUALITY-OF-SERVICE IN WIRELESS NETWORKS

Naveen Kumar Kakani, B.Tech(Hons), M.S.

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

August 2000

APPROVED:

Armin Mikler, Major Professor, Dept. of Computer Science
Roy T. Jacob, Committee Member and Chairman of the Department of Computer Science
Weiping Shi, Committee Member, Dept. of Computer Science
Paul Tarau, Committee Member, Dept. of Computer Science
Jianguo Liu, Committee Member, Dept. of Mathematics
C. Neal Tate, Dean of the Robert B. Toulouse School of Graduate Studies

Kakani, Naveen Kumar, Algorithms for Efficient Utilization of Wireless Bandwidth and to Provide Quality-of-Service in Wireless Networks, Doctor of Philosophy (Computer Science), August 2000. 113 pp., 2 tables, 16 illustrations, bibliography, 56 titles.

This thesis presents algorithms to utilize the wireless bandwidth efficiently and at the same time meet the quality of service (QoS) requirements of the users. In the proposed algorithms we present an adaptive frame structure based upon the airlink frame loss probability and control the admission of call requests into the system based upon the load on the system and the QoS requirements of the incoming call requests. The performance of the proposed algorithms is studied by developing analytical formulations and simulation experiments. Finally we present an admission control algorithm which uses an adaptive delay computation algorithm to compute the queuing delay for each class of traffic and adapts the service rate and the reliability in the estimates based upon the deviation in the expected and obtained performance. We study the performance of the call admission control algorithm by simulation experiments.

Simulation results for the adaptive frame structure algorithm show an improvement in the number of users in the system but there is a drop in the system throughput. In spite of the lower throughput, the adaptive frame structure algorithm has fewer QoS delay violations. The adaptive call admission control algorithm adapts the call dropping probability of different classes of traffic and optimizes the system performance w.r.t the number of calls dropped and the reliability in meeting the QoS promised when the call is admitted into the system.

ACKNOWLEDGMENTS

I am grateful to my advisor Dr. Armin Mikler for his encouragement, support and guidance at every stage of the work which constitutes this dissertation. I would like to take this opportunity to thank my committee members, Dr. Roy T. Jacob, Dr. Weiping Shi, Dr. Paul Tarau from the Department of Computer Science, and Dr. Jianguo Liu from Department of Mathematics for their review and suggestions for the improvement of this dissertation. As a special note I would like to thank Dr. Sajal K. Das from Department of Computer Enginering, UTA, for his help and encouragement in starting my thesis. I would also like to thank the Department of Computer Sciences at UNT for providing both computing and financial support during my tenure as a student.

Special thanks to my colleagues and friends at UNT, John Mayes, Paul Miller, and Vinay Balamuru for helping me at times when I ran into some problems with the my computer system. I would like to thank Dr. Allesandro Fabbri for his valuable suggestions to improve my thesis. I acknowledge the financial support from Texas Advanced Technology Program grant and from Northern Telecom (Nortel), Richardson, Texas. My internship with the Wireless Systems Engineering Department at Nortel provided me with valuable insights for improving the contents of this dissertation.

This task would not have been possible without the constant support and faith of my family and friends. Their "presence", love and encouragement have been a constant source of sustainance for me over the past few years.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOTATIONS[1]

CHAPTER 3

p_l: Airlink frameloss probability.

f_{QoS}: Quality of service function, product of the number of users served in each frame and the throughput of the system.

τ: Inter-packet delay.

$\mathcal{R}_{user}$: Data rate of service requests.

r: Number of retransmission minislots in a slot.

N: Number of slots in a frame.

N_1: Number of slots allocated to Mode-1 users in a frame.

N_2: Number of slots allocated to Mode-2 users in a frame.

M: Number of minislots in each slot.

k: Number of minislots assigned to user data transmission in each time slot assigned to Mode-1 users.

k': Maximum number of interleaved users who can be assigned to each Mode-2 slot.

R: Expected queue length for each minislot.

q_l: Probability of transmitting a frame successfully.

[1]Notations listed under each chapter may be referred in subsequent chapters but not repeated in their lists.

$\mathcal{R}_{min}$: Minimum data rate of the possible traffic in the system.

$\mathcal{R}_{slot}$: Data rate of the slot.

$\mathcal{R}_{header}$: The number of header bits expressed as data rate.

S_i^{Mode1}: State i in Mode-1 slot state transition diagram.

P_i^{Mode1}: Probability of being in state S_i^{Mode1} in Mode-1 state transition diagram.

k_{avg}: Expected number of mode-1 users in each timeslot. item [S_i^{Mode2}:] State i in Mode-2 slot state transition diagram.

P_i^{Mode2}: Probability of being in state S_i^{Mode2} in Mode-2 state transition diagram.

k'_{avg}: Expected number of frames during which Mode-2 slot is being used by a user.

$\hat{h}$: Ratio of the number of header bits to the number of bits in each timeslot.

η_{Mode1}: Throughput of Mode-1 slot.

η_{Mode2}: Throughput of Mode-2 slot.

Π_i^{Mode2}: Probability of successfully transmitting a packet at the $(i-1)$th retransmission attempt.

T_{Mode2}: Expected number of retransmissions for each packet.

η_{frame}: Throughput of the system.

f_N: Fraction of timeslots allocated to Mode-1 users.

U: Number of users whose data is sent in each frame.

f_{QoS}: Quality of Service function, product of the system throughput and the number of users in the system.

d_1: Number of data rates of Mode-1 traffic requests.

X_i^{Mode1}: Data rate i of Mode-1 request. ($1 \leq\ i\ \leq\ d_1$).

N_i^{Mode1}: Number of Mode-1 requests of data rate X_i^{Mode11}.

A_i^{Mode1}: Number of job requests of data rate X_i^{Mode1} assigned to a slot.

$Available_{Mode1}$: Available bandwidth in a Mode-1 slot.

d_2: Number of data rates of Mode-2 traffic requests.

X_i^{Mode2}: Data rate i of Mode-2 request. ($1 \leq\ i\ \leq\ d_2$).

N_i^{Mode2}: Number of Mode-2 requests of data rate X_i^{Mode12}.

A_i^{Mode2}: Number of job requests of data rate X_i^{Mode2} assigned to a slot.

$Available_{Mode2}$: Available bandwidth in a Mode-2 slot.

CHAPTER 4

F: Fractional change in airlink frame loss probability to trigger the slot reallocation algorithm.

k_{new}: Value of k for the new frameloss probability.

$N_{1_{new}}$: Computed value of N_1 for the new value of frame loss probability.

$N_{2_{new}}$: Computed value of N_2 for the new value of frame loss probability.

$\mathcal{MTU}$: Maximum Transmission Unit.

CHAPTER 5

N: Number of frames after which the service rate is adapted.

μ: Maximum service rate of the system.

Δ_μ: Fraction by which the service rate may be altered system at the end of the observation period.

μ_{avg}: Current service rate of the system.

E_N: Number of frames in error in the last set of N frames.

C_i: Class of traffic request.

D_i^{class}: Tolerable maximum delay of packets of class C_i.

$\hat{D}_i$: Estimate of the statistical delay if the new call is admitted.

D_p^i: Computed delay using M/M/1 model when the last call arrived.

$\hat{D}_p^i$: Statistical delay at the time of arrival of the last call.

λ_i: Arrival rate of call requests of class C_i.

λ_i^p: Peak rate of the new call new request of class C_i.

F: System utilization factor.

P_{drop}: Probability of dropping a call request even when it does not satisfy the call admission conditions A_1 or A_2.

λ: Call arrival rate to the system.

$Arrv_i$: Number of call arrivals of class C_i during the observation period N.

$Drop_i$: Number of call requests of class C_i dropped during the observation period N.

$Dviol_i$: Number of simulation cycles during which class C_i call requests had delay violations.

P^i_{drop}: Probability of dropping a call request of class C_i even when the call admission conditions are violated.

$Excess_i$: Excess delay than the maximum tolerable delay for class C_i call requests if the new call request is admitted.

$Excess_{sys}$: Excess service rate demanded by the system than the system can offer.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The fusion of computer and communication technologies has promised to herald the age of information super-highway over high speed communication wire-line and wireless networks. The ultimate goal is to enable a multitude of users from any place to access information at any time. In fact, the desire for ubiquitous access to information is expected to characterize entirely new types of information systems and technology as we move into the 21st century. The rapidly emerging wireless communication systems based on radio and infrared transmissions, the advent of technologies such as cellular mobile telephony, personal communication systems (PCS), wireless PBXs, wireless LANs (local area networks), and the promise of broadband ISDN (through gigabit networks) prelude a not-so-distant future realization of this goal. The driving force behind the field of mobile computing is also due to the commercial availability of hand-held, portable (e.g. laptop, palm-top) computers and personal digital assistants (PDAs) such as Apple Newton Message-Pad, and 3-Com Pilot. This field has the potential to dramatically change the society as workers are no longer away from their information sources and communication mechanisms. Wireless or cellular mobile computing has three essential components – communication, mobility and portability – which distinguish it from its wire-line counterpart. Wireless communication aspects deal with efficient bandwidth management and allocation of bandwidth to mobile applications (attempting to solve the fundamental problem in wireless domain – that of low bandwidth availability). Issues related to mobility include location

management for the mobile user, data management, loss of connections due to unreliable radio links and transmission security. Portability issues are concerned with the design of lightweight terminals with limited storage capacity and power consumption and also effective user interface design for them. Most of the existing algorithms have been proposed to fine tune the system to work optimally for voice applications. The next generation (3G+) wireless systems have to serve/support a large number of users with data applications which have different traffic characteristics [5, 6, 7] as compared to voice applications. As the current systems are fine tuned for voice applications, it is necessary to develop efficient bandwidth management schemes to handle different users traffic characteristics and at the same time increase the number of users supported by the system. As the number of users supported (along with QoS requirements) by the system increases more users will be subscribing to the system which will result in an increased revenue to the service provider. This thesis focuses on solving the problem of bandwidth allocation and selecting the users to be admitted into the system to meet the Quality of Service (QoS) requirements of users. In what follows, we propose a dynamic frame structure and statistical call admission control mechanism to control the amount and the type of traffic in the system.

1.1.1 QoS based frame structure

Within the European research program Advanced Communication Technologies and Services (ACTS), the project Future Radio Wideband Multiple Access System (FRAMES) [4] has been designed with an objective to define the radio interface for Universal Mobile Telecommunications Systems (UMTS).

In the United States, the wideband TDMA concept (frame with timeslots) based FRAMES multiple access (FMA) [4] without spreading i.e., only one user is allocated

to one timeslot, is proposed to be standardized as IS-136. The IS-136 protocol utilizes a sophisticated compression technology to optimize the system performance for voice applications. However, given the upsurge of data traffic in the next generation wireless networks, there is a need for new and more efficient bandwidth management protocols to enhance current systems which are primarily voice centric. The problem that we address is, finding an optimal frame structure for a given air-link frame loss rate and data rate of the applications. In this thesis we propose a novel frame structure in which the number of slots allocated to different types of traffic, the structure of a timeslot and the users admitted into the system depends upon the air-link frame loss rate. The objective is to operate the system at the point where the product of the system throughput and the number of users served by the system is maximized. We can increase the throughput of the system by reducing the number of overhead bits, this means that only a few users are assigned to each slot. So the objective of the system design is to incorporate more users in the system without dropping the throughput of the system.

1.1.2 QoS based call admission control

Many designs for integrated service networks offer a bounded delay packet delivery service to support real-time applications. To provide bounded delay service, networks must use proper admission control algorithm to regulate their load. Previous work on admission control mainly focused on algorithms that compute the worst case theoretical queuing delay to guarantee an absolute delay bound for all packets [8, 17] Traditional real-time applications provide a hard or absolute bound on the delay of each packet. Typically these are referred to as guaranteed service applications and existing solutions provide some form of bounded packet delay service. The ability

to achieve higher utilization and also meet service commitments depends crucially on the admission control algorithm [8, 17]. Conversely, the ability of an admission control algorithm to increase network utilization is ultimately constrained by service commitments the network makes.

Traditional approaches to admission control, like those used for guaranteed service, use a priori characterizations of sources to calculate the worst-case behavior of all the existing flows in addition to the new call which is waiting to be admitted by the system. Calculating the worst-case delays is generally very complex as, each application has different way of specifying its traffic characteristics and the worst case cannot be clearly defined. For example the specification of a voice application include average rate of transmission (V_{avg}) and the peak rate of transmission (V_{peak}), compare this specification to a data application which specifies token rate D_{avg}, peak rate D_{peak} and the bucket depth is D_{bucket} how to compute the worst case delays for these applications ? We can compute the delays by heuristics similar to the *effective bandwidth approach* proposed in [34] where computing the effective bandwidth itself is quite complex. In spite of the complexity of computing the worst-case delays the underlying admission control principle is conceptually simple: "if the system admits the new request for service, does this result in the worst-case behavior of the network and results in a violation of any delay bound of the existing request ?" For example consider the following scenario, a voice application is admitted into the system, based upon the resources allocated to it now it has a delay which is tolerable. If a new request is admitted some resources are allocated to the new request which means there will be an increase in delay of existing voice application If the resulting delay of the voice application is still within acceptable limits then the voice applications do not have delay violations, otherwise the new call should not be admitted. Moreover,

it is not possible to provide accurate statistical models for each individual flow as a user flow is an element of an infinite set of user patterns. Therefore, the *a priori* traffic characterizations negotiated by the user and the system at call arrival are loose upper bounds (worst case analysis). For further calculations for delay estimate of this call request, instead of basing the decision on the *a priori* characteristics negotiated by this call request and the system at call arrival time, the system measures the characteristics of the flow and the measured characteristics are used for the computations. We believe that measurement based admission control will play a key role in achieving high network utilization as we will be basing the decision based upon what the system has learned about a particluar call request from the time the call was admitted into the system until now. The measurement based admission control approach advocated in [24], [25] uses the a priori source characterizations only for incoming flows and uses measurements to characterize those flows that are in the system. Since the source behavior changes dynamically and the call admission depends upon the measurement based approach, the admission control can never provide the complete reliable delay bounds needed for guaranteed (or even probabilistic) service. Rather than using the measured data as it is the system can predict the expected performance based upon the performance of the system from a queuing model and the actual performance obtained. This is the essence of the proposed admission control algorithm.

1.1.3 Proposed Solutions

Rather than having a fixed frame structure change the frame structure with change in the airlink frame loss rate and hope to improve the system throughput and capacity. Given that it is difficult to characterize user traffic have the system adapt to the user traffic and make its future decisions from the knowledge it has acquired and control

the traffic in the system to reduce the rate of QoS violations of the users.

The proposed solution will require to understand the QoS requirements of different users the traffic generated by different applications and the way they are characterized with respect to data-rate and delay constraints. Users are assigned to different timeslots in a frame such that the QoS requirements of the users are not violated. Making a system learn the behavior of the traffic sources during their session and restricting the number of calls to the system eliminates the unreliability associated with a mechanism where calls are admitted based upon the *a priori* characterization of the sources. This results in systems that are robust, reliable and work efficiently for all traffic classes.

The bandwidth assignment problem is addressed in Chapter 2, the proposed framework for QoS based bandwidth assignment is presented in Chapter 3. Chapter 4 explains the algorithm for changing the frame structure with change in airlink frame loss rate and simulation experiments to show the performance of the new algorithm. The statistical call admission control algorithms, which predict the expected traffic based upon the knowledge acquired, are introduced in Chapter 5 along with the performance results from simulation experiments. The thesis is concluded with a summary of the results and observations in Chapter 6.

CHAPTER 2

THE BANDWIDTH ALLOCATION PROBLEM

In the literature, several types of access schemes have been proposed to better utilize the frequency spectrum [29, 30]. They include dynamic time division multiple access (D-TDMA), packet reservation multiple access (PRMA), resource auction multiple access (RAMA), and dynamic reservation multiple access (DRMA). In these schemes (as shown in Figure 2.1), a certain amount of bandwidth is reserved for sending reservation packets to reserve the bandwidth for a new traffic request or for detecting collisions. If there are multiple requests at the same time, user requests have to go through contention resolution mechanism. Multiple reservation requests received by the system merge together electrically and result in a collision packet. If a users request is accepted then the system sends a response with the timeslot numbers which the user can use to send data. This response is sent as a reservation packet.

The main difference in the existing access schemes is the amount of bandwidth allocated for making a reservation, to send user data, and the method in which a reservation is made. In PRMA each slot is marked as either reserved ($\mathcal{R}$) or available ($\mathcal{A}$). Slots marked as $\mathcal{A}$ are used by new users to send their request to the system. If the request is approved by the system then the slot is marked as $\mathcal{R}$. In D-TDMA and RAMA schemes, a part of the system bandwidth is dedicated for reservation slots (reservation slots are a fraction of each slot) and the rest is used to carry user data. In D-TDMA scheme, users send their request to use the system in the reservation slot. If the system has sufficient resources the system allocates timeslots to the user and informs the user of the reservation. Slot allocation is done by the system on

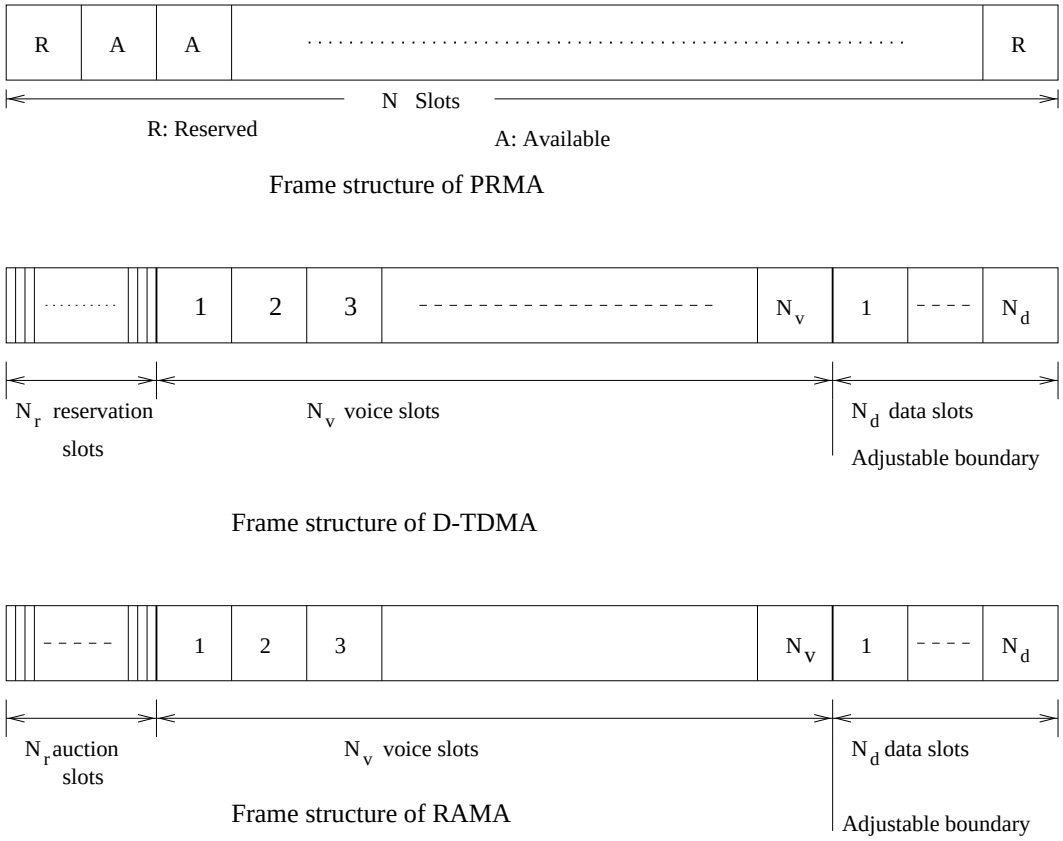

Figure 2.1: Existing frame structures

first come first serve basis. In RAMA users send their request to use the system with a certain incentive to the system. The system chooses the maximum incentive from the incentives given by the users and broadcasts the maximum incentive. This means that if a user wants to enter into the system he needs to give a higher incentive than what the system is broadcasting. In this way the system auctions the available slots. This process stops when there are no more incentives from users more than the bid which is broadcasted by the system. The user who has given the maximum bid is assigned time slots as per the requirement of his application. In D-TDMA and

RAMA scheme the number of slots assigned to voice users and data users change dynamically and they are arranged in sequence i.e., in a frame data slots follow voice slots. All the schemes exploit the concept of *fractional bandwidth* in the sense that they do not necessarily use the complete bandwidth corresponding to one time slot. This is because the reservation packet is much smaller than a data packet. The imminence of new services with a broad range of burstiness characteristics and their integration through statistical multiplexing has focussed on call admission as the primary instrument for rate-based congestion control. By preventing admission of an excessive number of calls or sources to the channel, call admission policies strike a balance between the grade of service and the efficient use of network resources. The effectiveness of statistical multiplexing which is the interleaving of frames during which a user is allowed to send data. This is done along with buffer sharing, which is allowing the users to store data if slot is not available [35], has long been proven and used in conventional communication networks to facilitate dynamic sharing of network resources. The statistical multiplexing system can be modeled as a multi-input single- server queuing system [36], which also proposes efficient computational algorithms to solve such models.

Within the European research program Advanced Communication Technologies and Services (ACTS), the project Future Radio Wideband Multiple Access System (FRAMES) [53] has been assessed with an objective to define the radio interface for Universal Mobile Telecommunications System (UMTS). During the first project year, a comprehensive evaluation of different multiple access technologies has been carried out [54]. As a result, FRAMES, multiple access (FMA) was selected. FMA consists of two modes:

FMA1-wideband time-division multiple access (TDMA).

- with spreading: Users are allocated timeslots along with a code because more than one user can be assigned to a time slot simultaneously. Users multiply their data with the code before they send it out. This will enable multiple users to send their data in the same frequency band.
- with out spreading: Only one user is allocated to each slot so users need not spread their data.

FMA2-wideband direct-sequence wideband code-division multiple access (CDMA). Users are assigned different channels and a code to spread their data. More than one user can be assigned to a channel.

Both options of FMA1 were contributed to the Special Mobile Group (SMG2) WB-TDMA and TDMA/CDMA concept groups respectively. In the United States, the wideband TDMA concept based on FMA1 without spreading, is proposed to be standardized as IS-136. The IS-136 protocol has provisions for a full-rate channel and a half-rate channel. The 48 Kbps $\frac{\pi}{4}$ DQPSK (differential quadrature phase shift keying) channel is divided into 40 msec frames, each of which is again subdivided into 6 slots. As per the standard a full-rate channel uses every third slot in the frame while a half-rate channel uses every sixth slot 2.2. Thus, a variable amount of bandwidth is allocated to the users which allows one user to transmit in 1 or 2 slots in one frame. The full-rate and half-rate channels support different data rate traffic to the same channel in IS-136. However, there is no change in the bandwidth allocated to a user as a function of change in the frameloss rate.

The two major differences [43] between wire-line and wireless networks are in *link characteristics* and the ability to handle *mobility* of users. The broadband wire-line network transmission links are characterized by high transmission rates (in the order

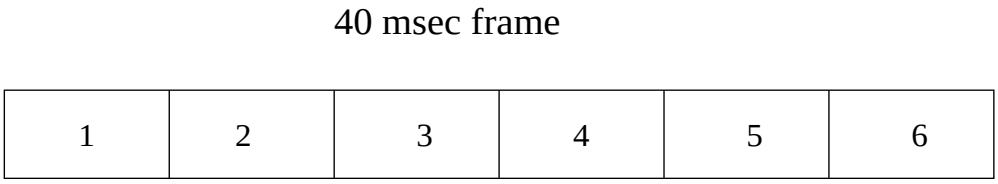

Figure 2.2: IS-136 Slot

of Gbps) and very low error rates (Ethernet standard IEEE 802.3 specifies a worst case bit error rate of 10^{-8} [2] and 10^{-18}-10^{-14} for fiber-optic links [3]). In contrast, wireless links have a much smaller transmission rate (Kbps-Mbps) and a significantly higher error rate (10^{-3} [1]). The most recent private wide area wireless data networks such as ARDIS or Mobitex offer a channel rate of about 8Kbps to 2Mbps and Motorola's Altair-II offers about 6 Mbps [44]. Additionally, wireless links experience losses due to multi-path dispersion and Rayleigh fading. The second major difference between the two networks is the user mobility. In wire-line networks, the user-network-interface (UNI) like an ethernet plug point, remains the same throughout the duration of a connection whereas the UNI in a wireless environment need not be fixed throughout the connection. During the duration of a call a wireless user can move from a region served by one base-station to a region served by another base-station i.e., he does not have a fixed UNI, i.e., the UNI is not static in nature.

Objectives of wireless system design:

- Support applications with varying QoS requirements.
- Optimize the grade of service offered by the system.

The slot allocation methods presented until now are *static* i.e., they do not have provision for dynamically changing the slot allocated to a user. To alleviate problems such as due to change in link characteristics in wireless links there should be a dynamic slot/slots allocation scheme to users based upon the airlink frame error rate rather than a static scheme. Moreover, the slot allocation algorithm should have a provision to handle users with a certain quality of service (QoS) requirements. Each user is guaranteed a QoS which takes into consideration the user's data rate requirements as well as the degradation of the system performance and the QoS offered to the user (grade of service) due to signal fading.

In summary this chapter provides an overview of the existing wireless access mechanisms, explains the differences between wireline and wireless networks, additional problems in building wireless networks and the requirements of slot allocation schemes to be able to be used to build wireless networks.

CHAPTER 3

FRAMEWORK FOR QoS BASED BANDWIDTH ALLOCATION ALGORITHM

This chapter presents a framework for bandwidth allocation in wireless networks based upon the datarate and interpacket delay requirements of the traffic. The proposed framework is designed to operate the system at an optimal point at which the product of the number of users and the throughput of the system is maximized. The point where the system is operated changes with a change in the airlink frame loss probability (p_l) and this is reflected by changing the frame structure.

The main idea behind the new bandwidth allocation algorithm is to reduce the time duration for which the system bandwidth is not in use. This is acheived by allocating one time slot to multiple users either by interleaving the users using the slot at frame level or split the slot into minislots and allocate minislots to users. In IS-136 protocol, a user can be allocated variable bandwidth with the provision of full-rate and half-rate channels, and a user transmitting in one slot in one frame also uses the same slot in the next frame. This strategy can ensure a definite inter-packet delay as the frame duration is constant and data is sent in every frame. However, given the various classes of services that can be requested, some users may not be concerned about the inter-packet delay as long as an acceptable average data rate of transmission can be guaranteed. So it is not a requirement that a user be allowed to send data in each frame.

With respect to the source we can broadly classify the user traffic as either constant bit rate (CBR) or variable bit rate (VBR) traffic. With reference to ATM systems CBR applications generate a constant cell-smooth traffic stream and expects that the

receiver will receive the stream with a small delay jitter and VBR referes to traffic that generates cells in bursts rather than in a smooth stream. The presence of the VBR traffic demands dynamic resource management techniques. Applying sophisticated compression algorithms, the 64 Kbps voice is compressed to 32, 24, 16, or even 8 Kbps of bandwidth. Silence suppression techniques are the major contributors to increased compression ratio. Due to the presence of compression algorithms, for CBR traffic probably the traffic does not require the entire timeslot. At the same time since we need to meet the interpacket delay all the users should transmit in each frame. One possible solution could be instead of multiplexing users at frame level each slot has to be split into minislots and users are assigned minislots. This ensures that each user sends data in each frame and at the same time his datarate requirements are met. The proposed algorithm coupled with the availability of additional bandwidth (allocated to wireless systems use) for the Wideband TDMA (WTDMA) schemes, can attempt to meet the QoS requirements of multimedia services as well. The throughput and the number of users served in each slot in each frame by the system is evaluated. The overall system parameter f_{QoS} is defined to be the product of the number of users served in each frame and the throughput of the system. The performance aspects $w.r.t$ the throughput, number of users served, and the deviation in the delay to send a packet is compared with the new algorithm and IS-136 protocol.

3.1 Classification of Traffic Requests

Based upon the inter-packet delays (τ) and the data rates ($\mathcal{R}_{user}$) of services, we divide the traffic into two broad classes denoted as Mode-1 and Mode-2. The traffic requests are characterized in Table 3.1.

Mode-1 Traffic Requests

Table 3.1: Traffic classes

Class	Mode-1	Mode-2
τ (Inter packet delay)	Provides a constant interpacket delay as demanded by the application to reduce the jitter	This class of traffic includes applications which are not effected by jitter
$\mathcal{R}_{user}$ (User data rate)	Guarantees minimum data rate as demanded by the application	Guarantees minimum data rate as demanded by the application
Applications	Voice, real time data	Email, fax

Mode-1 traffic users are accepted with an inter-packet delay constraint and the data rate of transmission. As an example, a voice call cannot tolerate a delay of more than $\tau = 250\ msec$ [56], and at the same time it requires certain data rate. To simplify the model, we assume that all requests in Mode-1 class cannot tolerate an inter-packet delay which is the time taken to send a fixed amount of data, no more than the time taken to send the same amount of data when the frame loss rate is zero. This means that all Mode-1 requests are transmitted during each frame. Moreover, based upon the datarate requirement a user is allocated the whole slot bandwidth or a fraction of the slot bandwidth. Figure 3.1 shows the slot structure for the new algorithm.

To reduce the increase in the interpacket delay due to packet loss (signal fading), a few minislots (fraction of timeslot) are used for retransmissions. A user can use any of these retransmission minislots to retransmit his lost packets. Since any user can use any of the retransmission minislots the system should broadcast the corresponding user ids of users who are using the retransmission minislots. One of the minislots in the forward link is used for broadcasting to all the users of the slot. Each slot has a retransmission list, upon loss of a frame all the packets sent in user minislots are appended to the retransmission list. In each frame, the first r packets from this list

are transmitted. If the frame is transmitted successfully the number of packets in the retransmission list reduces by r otherwise the length of the retransmission list increases by the number of user slots.

Mode-2 Traffic Requests

Mode-2 traffic demands a certain data rate based upon the application, and do not have any constraint on the inter-packet delay. Examples of the Mode-2 traffic are e-mail, ftp, fax, and so on. Each Mode-2 user transmits for the complete time slot duration allocated to him. Based upon the users data rate the user is interleaved with other users in using the time slot. As the inter-packet delay is not a concern, upon unsuccessful transmission of a packet a Mode-2 user sends the lost packet when he uses the timeslot at the next chance.

Figure 3.1 illustrates how users transmit in the time slots within a frame. The notations used in the figure are explained below.

- N = total # of slots in a frame
- N_1 = # of slots allocated to Mode-1 users
- N_2 = # of slots allocated to Mode-2 users
- M = # of minislots in each Mode-1 slot
- r = # of retransmission minislots in each Mode-1 slot
- k = # of minislots assigned for data transmission in each Mode-1 slot
- $k^{'}$ = maximum # of interleaved users who can be assigned to each Mode-2 slot
- R = Expected queue length of each minislot

The frame consists of N time slots out of which N_1 time slots are allocated to Mode-1 users and N_2 time slots are allocated to Mode-2 users. Each Mode-1 slot is divided into M minislots. Out of M minislots k minislots are allocated to transmit user data, r minislots are used to retransmit lost packets and one minislot is used to broadcast the ids of users currently occupying the retransmission minislots. Each Mode-2 slot is shared by $k^{'}$ Mode-2 users. A Mode-2 user who is using the slot in the current frame, say frame 1, uses the slot again after $k^{'}$ frames. Thus, we cannot allocate Mode-1 and Mode-2 users to a time slot simultaneously as Mode-1 users have to transmit in each timeslot.

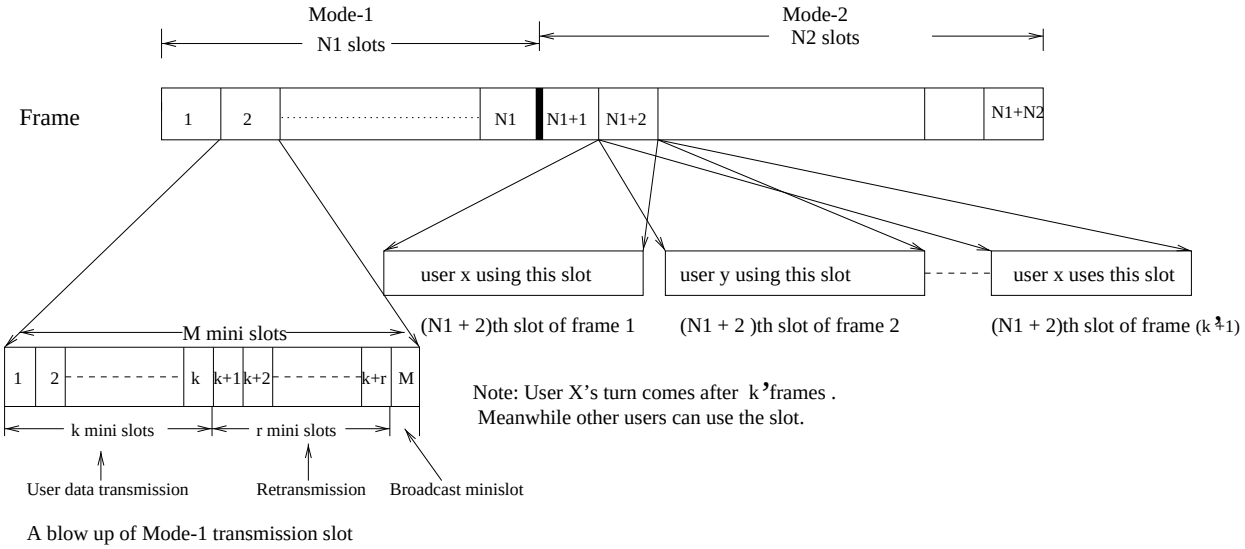

Figure 3.1: A snap shot of the frame structure as per our new algorithm

In summary,

- All the users in Mode-1 traffic transmit in each frame. Each user can transmit in a fraction of a time slot based on his data rate requirements.

- In Mode-2, only one of the k' users assigned to the time slot, transmits in that slot in a frame, and subsequently remains idle for k' frames. k' is determined by the data rate requirements of the users request.

3.1.1 Analysis of Mode-1 Request

For analysis purposes, we assume that a user who is allocated multiple minislots, transmits the header information in each minislot. This will ensure that the amount of header information is directly proportional to the number of minislots.

Since k minislots are allocated for user data transmission in each slot and The expected queue length of each minislot is R, the total number of retransmission minislots is $r = kR$. The value of R is obtained by modeling the retransmission list.

3.1.2 Number of Retransmissions

In this section a model is presented to obtain the length of the retransmission list. A user successful in transmitting a packet in a minislot in a frame, will proceed by transmitting a new packet in the same minislot in the next frame. If the frame is lost, the lost packets will be appended to the retransmission list. Since the retransmission is accomplished with the help of retransmission minislots, the user has the current minislot available to transmit the next packet.

Figure 3.2 shows the state diagram of the retransmission list for each slot. In this figure, p_l denotes the frame loss probability and q_l the probability of transmitting a frame successfully (i.e., $q_l = 1 - p_l$). Typically, the value of p_l depends upon the size of the data being transmitted and the forward error coding (FEC) technique used. In the state diagram as shown in Figure 3.2 a state represents the size of the retransmission list. State S_i represents a retransmission list of i packets. The state

transition algorithm is explained by the following example: let the list be in state S_0 i.e., there are zero packets to be retransmitted, Since each slot has k minislots for user data transmission, if there is any frame loss when the system is in state S_0 the next state of the retransmission list will be state $S_{l\times r}$ under the assumption that the ratio $\frac{k}{r} = l$ is an integer. Assuming the user sends a new packet in each frame, the probability of a transition from state S_0 to state $S_{l\times r}$ is p_l. In the next frame the lost packet are sent through the retransmission minislots and new packets are sent through user minislots. If this frame is lost, all the user packets are appended to the retransmission list. So the probability for a transition from state $S_{l\times r}$ to state $S_{2\times l\times r}$, denoted as $S_{l\times r} \longrightarrow S_{2\times l\times r}$, is given by p_l. If the state of the retransmission list is S_r, on successful transmission of the frame the retransmission list has zero packets. The probablity for the transition $S_r \longrightarrow S_0$ is q_l.

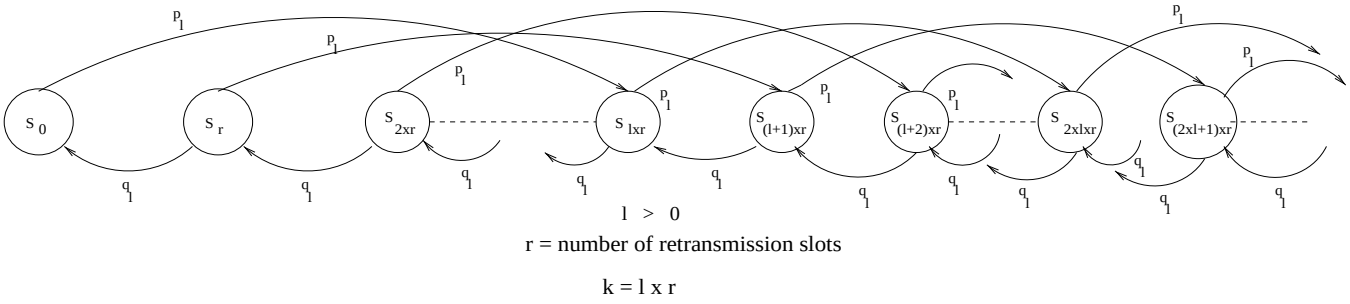

Figure 3.2: Modeling retransmission queue (list) for each user in Mode-1 traffic request

Evaluating the steady state probability, P_i, i.e., probability that the retransmission list has i packets is given by:

$$P_i = P_{i-k}\, p_l + P_{i+r}\, q_l \quad \text{for } i \geq k \quad \text{and } i = a\, r \quad \text{where } a \geq l$$

$$P_i = P_{i+r}\, q_l \quad \text{for } 0 < i < k \text{ and } i = a\, r \text{ where } 0 < a < l$$

$$P_0 = \left(\frac{q_l}{p_l}\right) P_r$$

The queue model shown in Figure 3.2 has a solution if we can limit the size of the queue. We analyze the system performance by restricting the queue length to be $100 * k$ and solving the above equations. The criterion for this assumption is the probability of each minislot having more than 100 packets in the retransmission list at any time is very small.

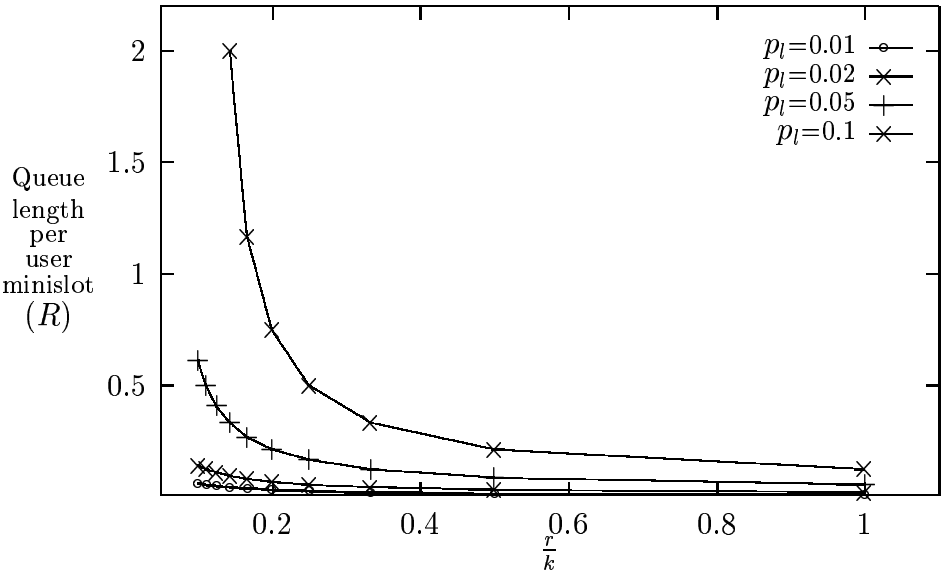

Figure 3.3: Retransmission queue length of each user minislot user

Figure 3.3 plots the queue length per minislot (i.e., R) by solving the steady state probabilities for varying frame loss probability (p_l) vs. the ratio $\frac{r}{k}$ of the number of retransmission slots to the slots assigned to user data.

Let $\mathcal{R}_{min}$ be the minimum data rate of the possible traffic in the system, and $\mathcal{R}_{header}$ the number of header bits expressed as data rate. The slot has a data rate of $\mathcal{R}_{slot}$, the total number of minislots is given by

$$M = \frac{\mathcal{R}_{slot}}{\mathcal{R}_{min} + \mathcal{R}_{header}}. \tag{3.1}$$

k out of these M minislots are assigned to users, $r = k \times R$ are assigned for retransmission, here R is the retransmission queue length of each user minislot and one of the minislot is assigned as a broadcast minislot. Thus $M = k + kR + 1$. The data in Figure 3.3 is approximated using a curve fitting software (Mathematica) as

$R = 4.69445\left(\frac{r}{k}\right)^2 - (0.5543 + 65.944\ p_l)\left(\frac{r}{k}\right) + 0.0417 + 6.37p_l + 190.739(p_l)^2$
Since $M = k + kR + 1 = k + r + 1$ substituting for R and r in terms of M and k leads to

$$M = K_1 \times \frac{(M-k-1)^2}{k} + K_2 \times k + K_3 \tag{3.2}$$

Where $K_1 = 4.69445$, $K_2 = 1.6 + 72.314125\ p_l + 190.74p_l^2$ and
$K_3 = -(M - 1)(0.5543 + 65.944125\ p_l) + 1$
Equation 3.2 is quadratic in k and the solution is given by

$$k = \frac{-\mathcal{B} + \sqrt{\mathcal{B}^2 + 4 \times \mathcal{A} \times \mathcal{C}}}{2 \times \mathcal{A}} \tag{3.3}$$

Where $\mathcal{A} = K_1 + K_2$, $\mathcal{B} = 2K_1 - 2MK_1 + K_3 - M$ and $\mathcal{C} = M^2K_1 - 2MK_1 + K_1$. Note that we are using the higher value of k (solution of a quadratic equation) since our objective is to incorporate a larger number of user-minislots rather than the number of retransmission-minislots. If $\mathcal{R}_{user}$ is the data rate of the application requested by a user, then the number of allocated minislots is $\lceil \frac{\mathcal{R}_{user}}{\mathcal{R}_{min}} \rceil$.

At the end of each frame if the users of minislots terminate their calls the system computes the amount of bandwidth available and allocates new user requests to the available bandwidth.

3.1.3 Expected Number of Users in Each Mode-1 Slot:

Figure 3.4 shows the state diagram of a Mode-1 request, where a state S_i^{M1} $1 \leq i \leq k$, denotes the number of minislots in use in each slot.

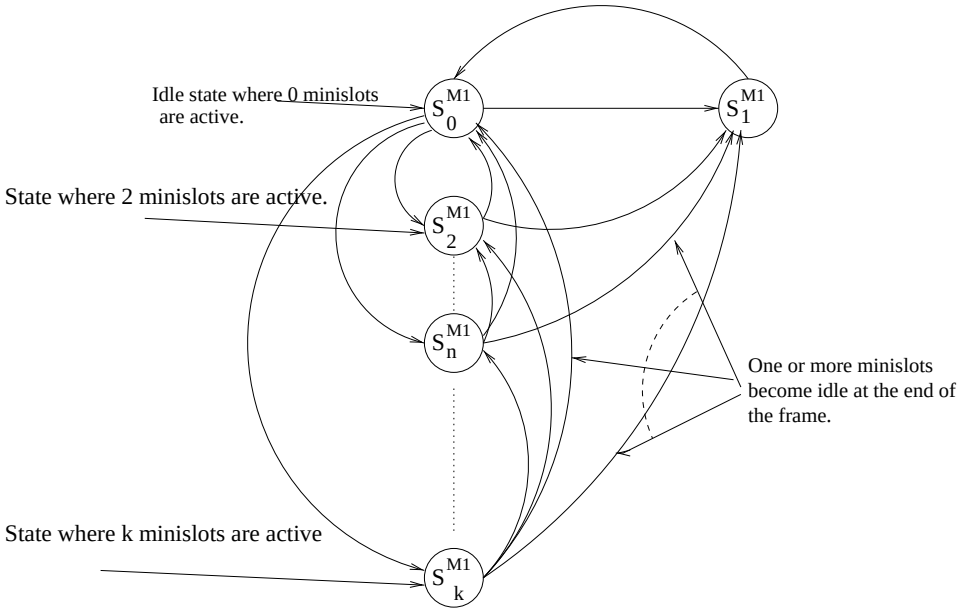

Figure 3.4: Possible states of the slot for Mode-1 traffic.

In the state diagram shown in Figure 3.4, since each user can be allocated multiple number of minislots, if a user terminates his call, one or more minislots may become idle. Moreover and at the end of a frame more than one user can terminate their call. If S_n^{M1} is the current state of the slot, the next state can be any one of the states S_{n-1}^{M1}, S_{n-2}^{M1} $\cdots$ S_1^{M1}, S_0^{M1} for all $0 < n \leq k$.

Denoting P_i^{Mode1} as the steady state probability of being in state S_i^{M1}, the expected number of active slots is given by

$$k_{avg} = \sum_{i=1}^{k} i \times P_i^{Mode1} \tag{3.4}$$

To evalute the steady state probablity we need to compute the transition probability from one state to another state. Given that a slot can have have many combinations of requests to fill up the available bandwidth the transition probabilites are not computable. We compute the expected number of minislots being used from simulation experiments. The simulation setup to evaluate the expected number is explained in Chapter 4. For analysis purposes we assume that we have an expected value of the number of active slots.

3.1.4 Analysis of Mode-2 Request

The requests assigned to Mode-2 slots require datarate based upon the application the are running. Since no separate slots are allocated for retransmission, the user has to retransmit a lost packet at the earliest opportunity.

Expected Number of Users in Each Mode-2 Slot:

For Mode-2 requests, each user is allowed to use a time slot in a frame after a certain (fixed) number of frames, $\lceil \frac{\mathcal{R}_{slot} - \mathcal{R}_{header}}{\mathcal{R}_{user}} \rceil$, $\mathcal{R}_{user}$ is the data rate of the user request. The maximum number of interleaved frames after which a user uses the slot is denoted by $k^{'}$ and it is given by:

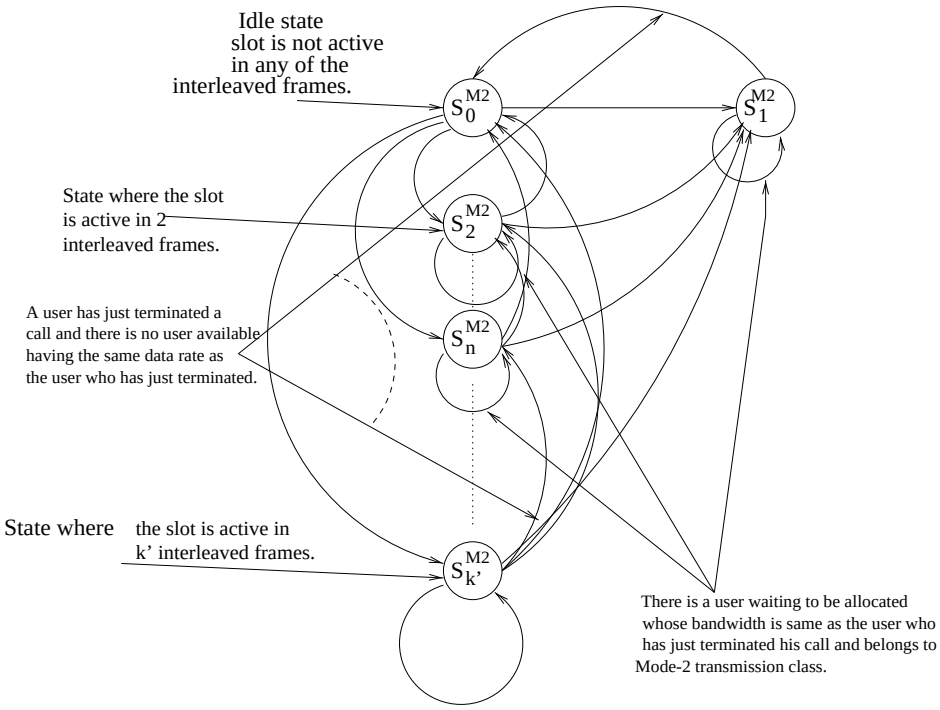

Figure 3.5: Possible states of the slot for Mode-2 traffic

$$k' = \frac{\mathcal{R}_{slot} - \mathcal{R}_{header}}{\mathcal{R}_{min}}. \tag{3.5}$$

Figure 3.5 depicts the state diagram for a Mode-2 request. The state S_i^{M2}, for $0 \leq i \leq k'$, denotes the number of interleaved frames that have the Mode-2 slot transmitting useful data. Once a user terminates a call, compute the available bandwidth and allocate new requests to that slot.

Consider the state of the slot S_n^{M2} which denotes that there are n interleaved frames that are active. Now if a user assigned to the slot terminates the call, the state of the slot can be S_n^{M2}, S_{n-1}^{M2}, S_{n-2}^{M2} $\cdots$ S_1^{M2}, S_0^{M2} where $0 < n \leq k'$. We

compute the available bandwidth in the slot and assign new users to the slot.

By evaluating the steady state probabilities numerically, the expected number of active interleaved frames is obtained as

$$k'_{avg} = \sum_{i=1}^{k'} i \times P_i^{Mode2}. \tag{3.6}$$

To compute the steady state probabilites numerically we need to compute the transition probablity from one state to other possible states. Since this is not computable as the number of combinations of job requests that make up for the number of interleaved frames being used are large, we obtain the expected number of active interleaved frames from simulation experiments.

3.2 Performance Analysis of the Overall System

The rationale for classifying job requests into Mode-1 and Mode-2, is to find a mixture of jobs to be assigned to timeslots in the frame such that the idle period of the slot is minimized and we have have a higher revenue. However, we may sacrifice the *throughput* of the system which is defined as the ratio of the number of data bits transmitted in one frame to the total number of bits (including the header bits). The degradation in the throughput is due to assigning multiple users to a timeslot which means we have more overhead bits and compare this to the situation in which one user is assigned to a timeslot (no additional overheads). Thus, from the viewpoint of the system performance, there is a tradeoff between the throughput and the *capacity* (number of users served) of the system. Consider Mode-1 jobs, as the number of users (minislots in a slot) is increased (additional revenue) for a few more users there

is no loss in the throughput because we are reducing the idle time of the user but beyond certain limit there is a drop in the throughput. The objective in this situation is to find the point where the loss of throughput is not more than the gains due to additional users which is a direct function of the revenue earned.

For Mode-2 jobs, only one user is allowed to transmit in each frame as only one header information is transmitted in each time slot. Although throughput is not sacrificed, the inter-packet delay is increased as more Mode-2 users are added. In this analysis we are not concerned about the inter-packet delay for a Mode-2 call request (property of Mode-2 requests).

Throughput Measurement

For each Mode-1 request, k minislots are assigned to users. So the number of retransmission minislots is kR and there is one broadcast minislot. Let D be the bandwidth or the total number of bits that can be sent in one time slot duration. Let h be the header length for each of the minislots and let $\hat{h} = \frac{h}{D}$.

Theorem 1: The throughput of a Mode-1 slot is given by

$$\eta_{Mode1} = \frac{\left(\frac{\mathcal{R}_{slot}}{k+kR+1} - \mathcal{R}_{header}\right)}{\frac{\mathcal{R}_{slot}}{k+kR+1} + \frac{\mathcal{R}_{slot}(kR+1)}{k(k+kR+1)}} \left(\frac{k_{avg}}{k}\right) \tag{3.7}$$

Proof: For an average case analysis, the number of retransmission minislots should be sufficient to send the data for each user. Thus, distributing the bandwidth uniformly over all the k user minislots in the slot, the effective bandwidth per minislot is the sum of the bandwidth of each minislot ($\frac{\mathcal{R}_{slot}}{k+kR+1}$) and the fraction of

the retransmission minislots bandwidth for each minislot ($\frac{\mathcal{R}_{slot}(kR+1)}{k(k+kR+1)}$). The bandwidth for data transmission is $\frac{\mathcal{R}_{slot}}{k+kR+1} - \mathcal{R}_{header}$. On an average, only k_{avg} of the possible k minislots are active. Thus the throughput of the Mode-1 request is given by

$$\eta_{Mode1} = \frac{(\frac{\mathcal{R}_{slot}}{k+kR+1} - \mathcal{R}_{header})}{\frac{\mathcal{R}_{slot}}{k+kR+1} + \frac{\mathcal{R}_{slot}(kR+1)}{k(k+kR+1)}} \left(\frac{k_{avg}}{k}\right). \tag{3.8}$$

For the Mode-2 users, let us assume that k' users are assigned to each slot. Since each user is allowed to use the complete bandwidth of $\mathcal{R}_{slot}$, the efficiency of all the users using the channel depends upon the average number of transmissions needed for one packet. If there is no signal loss, the efficiency of each slot is given by $\frac{\mathcal{R}_{slot} - \mathcal{R}_{header}}{\mathcal{R}_{slot}}$. Upon loss of frame the data sent in the previous frame has to be retransmitted. The efficiency of a Mode-2 slot depends upon the expected number of transmissions needed for successful transmission of a packet.

The probability that a Mode-2 user can transmit a data packet is given by $q_l = \Pi_1^{Mode2} = (1 - p_l)$, here p_l is the frame loss probability. The probability of success in the first retransmission (second transmission) attempt is $\Pi_2^{Mode2} = p_l(1 - p_l)$. In general, the probability of occurrence of the event of sending the packet in the n^{th} attempt is given by $\Pi_n^{Mode2} = p_l^{n-1}(1 - p_l)$. Therefore, the estimated number for transmissions for each packet is

$$\begin{aligned} T_{Mode2} &= \sum_{i=1}^{\infty} i * \Pi_i^{Mode2} = (1 - p_l) \sum_{i=1}^{\infty} i p_l{}^{i-1} \\ &= \frac{1}{1 - p_l}. \end{aligned} \tag{3.9}$$

Each of the $k^{'}$ users requires T_{Mode2} number of time slots to transmit his packet of data. Each transmission uses the full bandwidth $\mathcal{R}_{slot}$, out of which $\mathcal{R}_{slot} - \mathcal{R}_{header}$ contains the useful data. On an average, only $k^{'}_{avg}$ of the possible $k^{'}$ users are assigned to each slot. Thus the throughput of Mode-2 slot is given by

$$\eta_{Mode2} = \frac{\mathcal{R}_{slot} - \mathcal{R}_{header}}{T_{Mode2} * \mathcal{R}_{slot}} \left(\frac{k^{'}_{avg}}{k^{'}} \right). \tag{3.10}$$

Let N_1 slots be assigned to Mode-1 users and N_2 slots to Mode-2 users, the total number of slots in a frame is $N = N_1 + N_2$, which is a constant. The fraction of the header bits in minislot to the total number of bits in each slot is $\hat{h}$, the number of user minislots in each Mode-1 slot is k and R is the retransmission queue length for each user minislot, $f_N = \frac{N_1}{N_1 + N_2}$. The throughput of a frame is the average of the throughput of Mode-1 slots and Mode-2 slots in the frame and it is given by:

$$\begin{aligned} \eta_{frame} &= \frac{N_1 * \eta_{Mode1} + N_2 * \eta_{Mode2}}{N_1 + N_2} \\ &= f_N \left[\frac{1 - \hat{h}(k(1 + R) + 1)}{k_{avg}(k(1 + R) + 1)} \right] \\ &\quad + (1 - f_N)(1 - \hat{h})(1 - p_l) \left(\frac{k^{'}_{avg}}{k^{'}} \right) \end{aligned} \tag{3.11}$$

Given a fixed air link frame loss rate (p_l) the retransmission queue length of each user minislot (R), the number of user minislots (k) in each slot, average number of minislots that are in use in each Mode-1 slot (k_{avg}), average number of interleaved frames ($k^{'}_{avg}$) are fixed. The only parameter which can change the throughput of the system is the fraction f_N.

Number of Users

The total number of users served by the system in each frame is nothing but the average number of Mode-1 users and the average number of Mode-2 users in each frame. The number of users is given by

$$U \;=\; N_1 k_{avg} \;+\; N_2 \left(\frac{k^{'}_{avg}}{k^{'}}\right)$$

Quality of Service

We define the *quality of service* function, f_{QoS}, as the product of the number of users in the system per slot per frame and the system throughput. Thus, $f_{QoS} = \eta_{frame}\left(\frac{U}{N}\right)$

$$\begin{aligned} f_{QoS} \;=\; & \left[f_N\left(\frac{k_{avg}(1-\hat{h}(k(1+R)+1))}{k(1+R)+1}\right) + (1-f_N)(1-\hat{h})(1-p_l)\left(\frac{k^{'}_{avg}}{k^{'}}\right)\right] * \\ & \left[f_N k_{avg} + (1-f_N)\left(\frac{k^{'}_{avg}}{k^{'}}\right)\right] \end{aligned} \tag{3.12}$$

In this formulation, the only parameter that can be tuned for an optimal performance is $f_N = \frac{N_1}{N_1+N_2}$ which is obtained by a standard optimization technique [33]. Differentiating Equation (3.12) w.r.t. f_N leads to

$$f_N \;=\; -\left[\frac{k^{'}_{avg}(\eta_{Mode1} \;-\; \eta_{Mode2}) \;+\; (k_{avg} * k^{'} \;-\; k^{'}_{avg})\eta_{Mode2}}{2(\eta_{Mode1} \;-\; \eta_{Mode2})(k_{avg} * k^{'} \;-\; k^{'}_{avg})}\right] \tag{3.13}$$

for which the function f_{QoS} is optimal.

To maximize f_{QoS}, we take the second derivative of Equation (3.12) w.r.t. f_N which yields the following condition

$$C1: \ 2(\eta_{Mode1} - \eta_{Mode2})(k_{avg} * k^{'} - k^{'}_{avg}) \leq 0. \tag{3.14}$$

Recalling that f_N satisfies $0 \leq f_N \leq 1$, from condition $C1$ we derive two other conditions as follows.

$$C2: \ \frac{\eta_{Mode1}}{\eta_{Mode2}} \leq 2 - \frac{k^{'} * k_{avg}}{k^{'}_{avg}} \tag{3.15}$$

$$C3: \ \frac{\eta_{Mode1}}{\eta_{Mode2}} \geq \frac{4k^{'}_{avg} - 3k^{'} * k_{avg}}{3k^{'}_{avg} - 2k^{'} * k_{avg}} \tag{3.16}$$

For f_N (as per Equation 3.13) to be the optimal point of operation conditions $C1 - C3$ have to be satisfied.

3.3 Proposed Slot Assignment Algorithm

Equation 3.13 computes the fraction of slots in a frame that are to be assigned to Mode-1 requests and the fraction of slots that are assigned to Mode-2 requests are given by $1 - f_N$. Equation 3.3 computes the number of minislots to be assigned to users in each Mode-1 slot and the number of retransmission minislots are $M - k - 1$. Based upon the derived system parameters, from the available job requests the system has to determine which job requests to be admitted into the system and which ones to be dropped. For each slot there is certain amount of bandwidth that is available to be allocated to user requests. Apart from this for a Mode-1 slot or a Mode-2 slot there is an upper limit on the number of users who can be assigned to each slot. The objective in choose the users request to minimize the amount of bandwidth wasted and at the same time maximize the number of users who are allocated to the system.

More number of users more revenue. The solution is based on a linear programming formulation. In the following section we develop this linear programming formulation to make a choice of job requests to be served.

3.3.1 Linear Programming Formulation for Mode-1 Slot

Let d_1 denote the number of possible data rates which can be expected from the Mode-1 traffic. These data rates are denoted as X_i^{Mode1} where $1 \leq i \leq d_1$. Let the number of requests waiting to be assigned to these data rates be denoted by N_i^{Mode1}, let the number of jobs of each data rate assigned to a slot be denoted by A_i^{Mode1} where $1 \leq i \leq d_1$. Since the number of minislots assigned to users in each slot is known, the available data bandwidth is $Available_{Mode1}$ (which is equal to $k\,\mathcal{R}_{min}$ if the slot has no users assigned to it) which is the bandwidth available to transmit data after the header information for all users in that slot is transmitted. In order to minimize the amount of data bandwidth wasted, we find an optimal combination of user requests such that the bandwidth required is very close to the available bandwidth. The linear programming formulation for determining the combination of data rates for one slot allocation for Mode-1 jobs can be written as

$Minimize\ \mathcal{F} \geq 0;$

$\mathcal{F} = \lfloor \frac{Available_{Mode-1}}{\mathcal{R}_{min}} \rfloor - \sum_{i=1}^{d_1} A_i^{Mode1}$

subject to the following constraints

$Available_{Mode1} - \sum_{i=1}^{d_1} X_i^{Mode1} A_i^{Mode1} \geq 0$

$A_i^{Mode1} \leq N_i^{Mode1}$ and $A_i^{Mode1} \geq 0$ for $1 \leq i \leq d_1$.

At the end of each frame duration the number of minislots that are idle are computed and the linear programming formulation is called to allocate job requests to the available minislots.

3.3.2 Linear Programming Formulation for Mode-2 Slot

The formulation for the Mode-2 slot is similar except that the available bandwidth for transmission by a user is $Available_{Mode2} = \mathcal{R}_{slot} - \mathcal{R}_{header}$ when there is no user assigned to the slot. This is because the complete slot bandwidth corresponding to each frame is occupied by one user only. For Mode-2 jobs, let d_2 denote the number of possible data rates which are denoted as X_i^{Mode2}. Let the number of job requests for each data rate waiting to be allocated be N_i^{Mode2}, and let the number of jobs of each data rate assigned to a slot be denoted as A_i^{Mode2} where $1 \leq i \leq d_2$.

$Minimize\ \mathcal{F} \geq 0;$

$\mathcal{F} = \lfloor \frac{Available_{Mode2}}{\mathcal{R}_{min}} \rfloor - \sum_{i=1}^{d_2} A_i^{Mode2};$

subject to the following constraints

$Available_{Mode2} - \sum_{i=1}^{d_2} X_i^{Mode2} A_i^{Mode2} \geq 0$

$A_i^{Mode2} \leq N_i^{Mode2}$ and $A_i^{Mode2} \geq 0$ for $1 \leq i \leq d_2$.

At the end of each frame duration the available bandwidth is computed and the linear programming formulation is called to allocate job requests to the available minislots.

CHAPTER 4

DYNAMIC QOS BASED BANDWIDTH ALLOCATION ALGORITHM

Chapter 3 presented the design of the frame structure for a fixed value of the airlink frame loss probability(p_l). In existing networks, the airlink frame loss rate fluctuates over the range 0.01 to 0.15 with a mean of 0.03. If the frame structure is fixed, with a change in airlink frame loss probability the system will deviate from the optimal point of operation (maximum value of f_{QoS}). As shown in Chapter 3 for a fixed value of airlink frame loss rate (p_l) the parameter which can tune the system performance is f_N, which is the ratio of the number of slots allocated to Mode-1 requests to the total number of slots. To operate the system at the point where the performance is optimal with a change in airlink frame loss rate change the frame structure.

As shown in Chapter 3, the number (k) of minislots in each slot that are assigned to users is dependent upon the airlink frame loss probability and it is given by Equation 3.3. In the proposed channel assignment algorithm described below, compute k (number of minislots in a slot) and f_N (ratio of number of Mode-1 slots to the total number of slots in a frame) with the updated value of airlink frame loss probability. An increase in f_N, demands that some slots which are allocated to Mode-2 should now be allocated to Mode-1 users and a decrease in f_N demands that some slots (given by the change in f_N) allocated to Mode-1 users should now be allocated to Mode-2 users. This necessitates the reallocation of users of a particular class to another slots which carry traffic of the same class, to the empty slots obtained allocate new job requests. The proposed slot reassignment algorithm works on a greedy mechanism and it is described below.

4.1 Slot Reassignment Algorithm

Figure 4.1 shows the flow chart of the proposed slot reallocation algorithm. The channel reallocation algorithm is executed when the airlink frame loss probability changes by a fraction $\pm F$. For the new value of the frame loss probability, compute the value of k and N_1, denote them by k_{new} and $N_{1_{new}}$, now $N_{2_{new}} = N - N_{1_{new}}$. If $N_{1_{new}}$ is more than N_1 *i.e.,* some Mode-2 slots have to be migrated to Mode-1 slots. Free $(N_{1_{new}} - N_1)$ number of Mode-2 slots, *i.e.,* adjust the current Mode-2 traffic which is using N_2 number of slots in $N_{2_{new}}$ slots and allocate new Mode-1 users to the free slots. If $N_{1_{new}}$ is less than N_1 free $(N_1 - N_{1_{new}})$ number of Mode-1 slots and allocate new Mode-2 users to the free slots. Figure 4.2 shows the flow chart to free Mode-1 slots. The flow diagram to free Mode-2 slots is similar to the one shown in Figure 4.2, except that instead of freeing $N_1 - N_{1_{new}}$ Mode-1 slots free $N_{1_{new}} - N_1$ Mode-2 slots. To select the slots of a particular class to be freed employ a greedy strategy. Free $(N_{1_{new}} - N_1)$ number of Mode-2 slots or $(N_1 - N_{1_{new}})$ number of Mode-1 slots which have the least amount of bandwidth in use. Detailed algorithm is presented below:

1) Sort the Mode-1 or Mode-2 slots which ever have to be reassigned in increasing order of available bandwidth.

2) From the sorted slot array choose the last $N_1 - N_{1_{new}}$ slots for reassignment if Mode-1 slots have to be reassigned or choose the last $N_{1_{new}} - N_1$ slots if Mode-2 slots have to be reassigned.

3) For each slot which has to be reassigned, if the calls in the slot can be adjusted in the slots of the same class which are not going to be reassigned, decrement

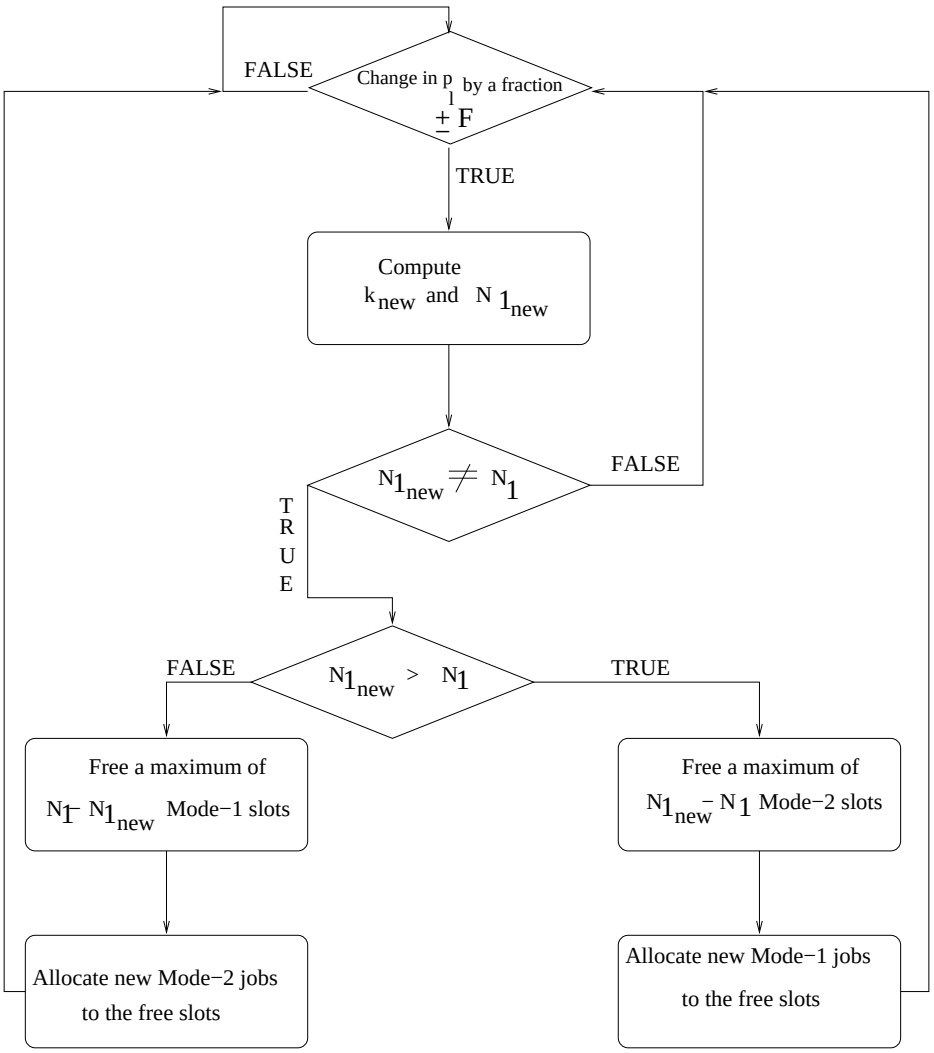

Figure 4.1: Slot Reallocation Algorithm

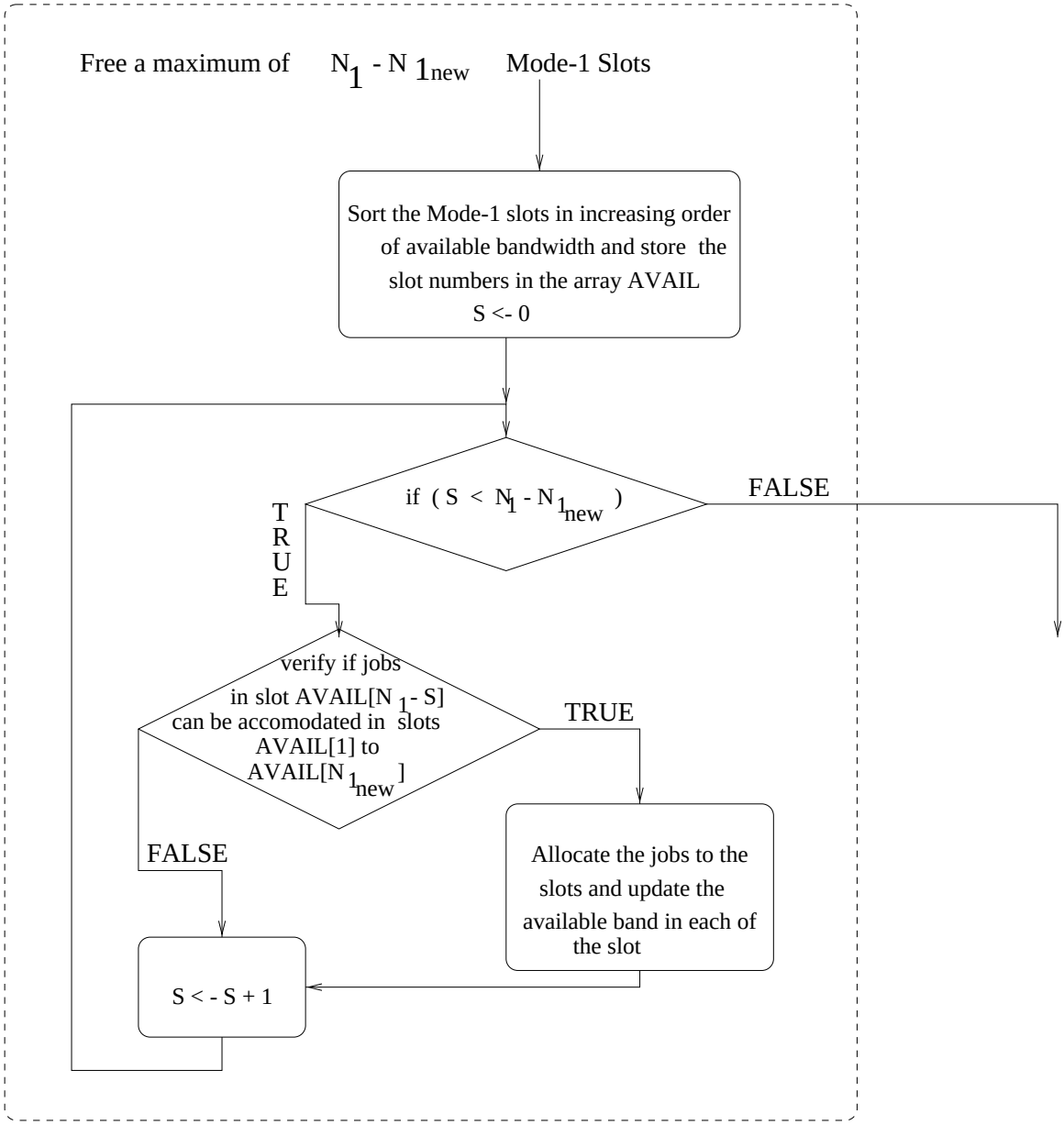

Figure 4.2: Algorithm to free Mode-1 slots

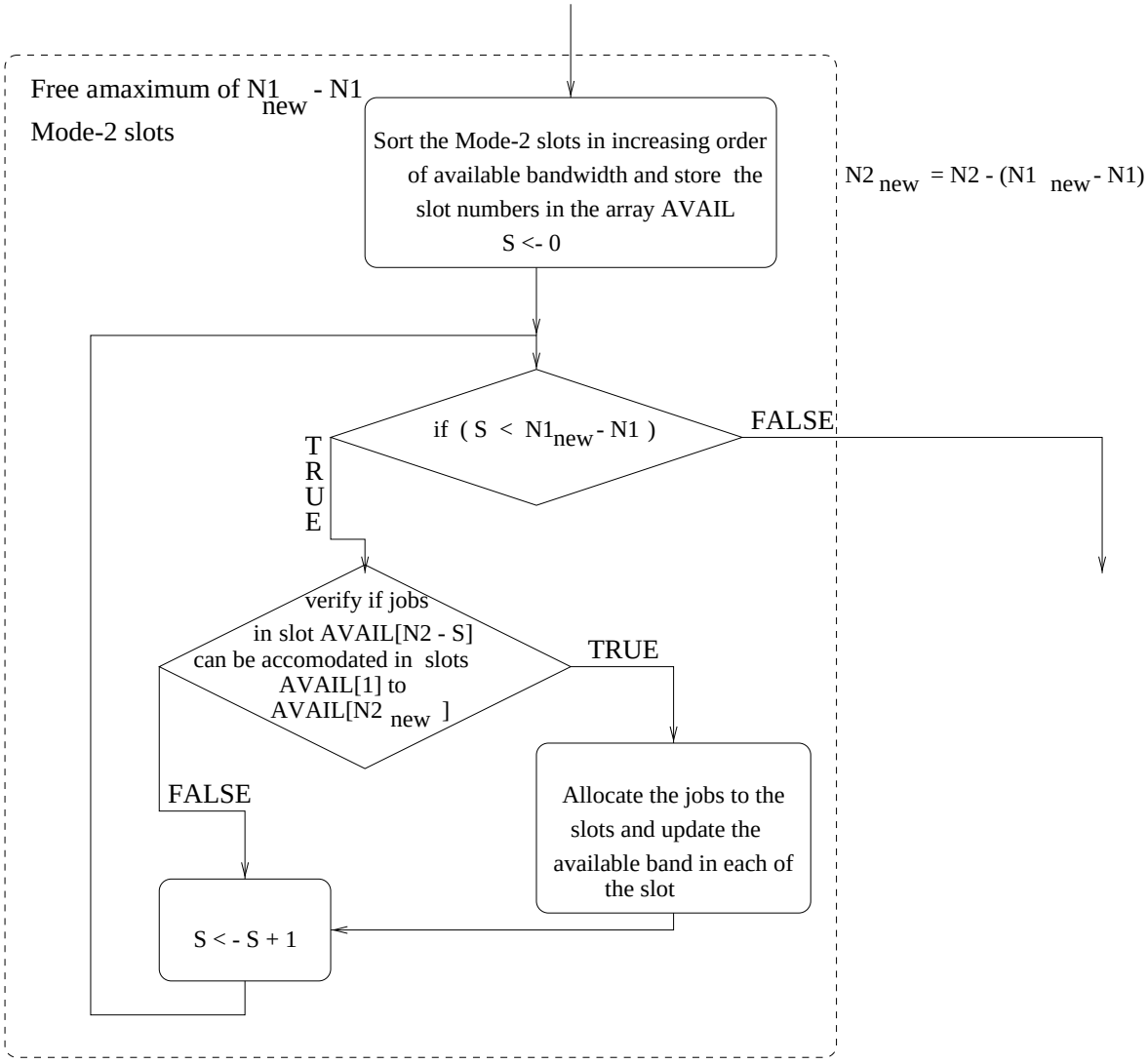

Figure 4.3: Algorithm to free Mode-2 slots

the available bandwidth of the slots to which the calls will be assigned by the bandwidth of the calls which will be assigned to that slot and assign the call to that slot. Allocate new job requests to the freed slot. If the calls cannot be adjusted then leave the slots as they are.

Due to the greedy property in the reassignment algorithm it is not guaranteed that the slots which are chosen for reallocation to be in a sequence. However, since the frame structure shown in Figure 3.1 is designed to have Mode-2 slots after Mode-1 slots, after the reassignment if there are slots which are out of sequence they need to be swapped to maintain the sequence.

4.2 Dynamic Slot Assignment Algorithm

The slot assignment algorithm is as follows:

1) Compute the number of minislots to be allocated to users of each Mode-1 slot from Equation (3.3).

2) Compute the number of interleaved frames for each Mode-2 slot from Equation (3.5).

3) Maximize the f_{QoS} formulation and obtain the fraction of slots to be allocated to Mode-1 requests from Equation (3.13).

4) Allocate requests to Mode-1 and Mode-2 slots by executing their linear programming formulations (as discussed in Chapter 3).

5) Upon loss of frame, all Mode-1 packets of a slot are appended to the retransmission list of that slot. For a Mode-2 slot, a user retransmits the packet in the next opportunity.

6) At the end of each frame compute the available bandwidth in each slot and assign new job requests to the available bandwidth by executing the linear programming formulation.

7) If the airlink frame loss probability changes by a fraction $\pm F$, execute the algorithm to reallocate the slots (Figure 4.1).

The reassignment algorithm involves sorting of arrays and reassigning calls to other slots, this requires computation power so the reassignment algorithm is not executed after every frame. Instead it is executed for a change in p_l by $\pm F$. Moreover, given the characteristics of wireless links, if the reassignment algorithm is executed for a change in p_l then it is most likely that the system would be busy at the end of each frame computing a new frame structure.

4.3 Simulation Experiments

For simulating the proposed slot assignment algorithm we have considered six data rates of traffic for both Mode-1 and Mode-2 class of requests. These data rates are 2 Kbps, 4 Kbps, 8 Kbps, 16 Kbps, 24 Kpbs, and 32 Kbps. We assume these data rates because they cover a wide range of applications, voice at 8 Kbps, email, fax at 2 Kbps to 4 Kbps, real time applications at 32 Kbps. The frame is generated with $N = 50$ slots [40] and the bandwidth of each slot is 48.6 Kbps [55]. The proposed simulation model is as shown in Figure 4.4.

Each simulation cycle corresponds to one frame duration. During each simulation cycle there can be at most four call requests on each channel and there are eight channels in the system [39, 40]. Hence, at the end of each frame, there can be at most 32 call requests. All the call requests have a service rate of μ which is set to

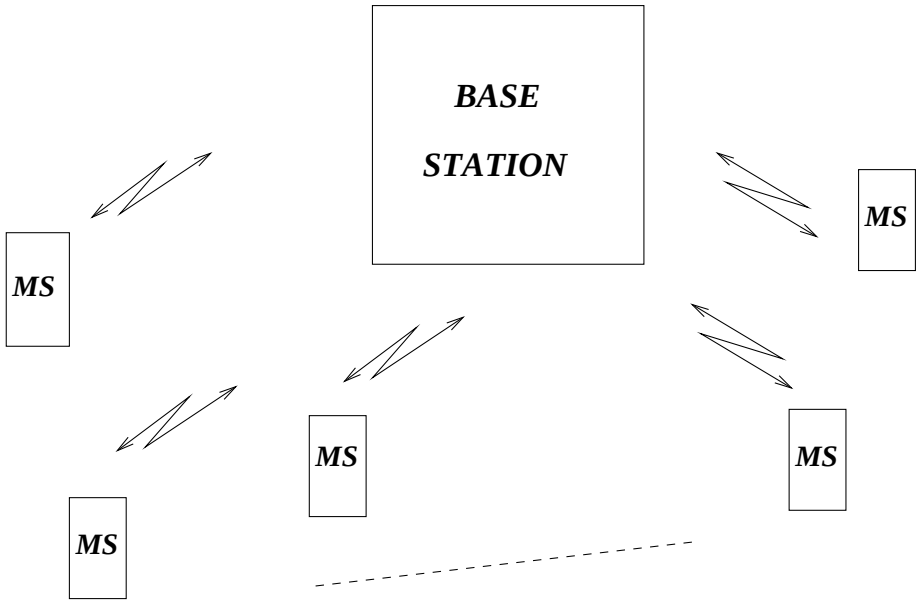

Figure 4.4: Simulation Model

0.01, the overhead bits in each slot and minislots ($\hat{h}$) is set to 0.05 i.e., the number of overhead bits is 5% of the slot bandwidth (5% as in IS-136 system [55]). We run the simulation experiments for varying airlink frame loss probability.

4.3.1 Computing k_{avg} and k'_{avg}

In Chapter 3 Equation 3.13 is obtained assuming we know the average number of minislots in a Mode-1 slot that carry user data in each frame (k_{avg}) and the average number of interleaved frames that carry user information in each Mode-2 slot (k'_{avg}). The state transition models of Mode-1 and Mode-2 slots are shown in Figures 3.4 and 3.5. We compute k_{avg} and k'_{avg} by simulation experiments. Run simulation experiments with a constant value of k_{avg} and k'_{avg}. Initialize k_{avg} and k'_{avg} to 0.1 and increment them in steps of 0.1. Results are tabulated for the value of k_{avg} and k'_{avg}

Table 4.1: Optimal Values of k_{avg} and k'_{avg}

p_l	k_{avg}	k'_{avg}	f_{QoS}
0.01	0.6	0.5	2.633
0.02	0.7	0.6	6.637
0.03	0.7	0.6	2.075
0.04	0.6	0.5	2.068
0.05	0.6	0.5	2.060
0.06	0.6	0.5	2.052
0.07	0.7	0.5	2.585
0.08	0.7	0.5	2.577
0.09	0.7	0.5	2.577
0.1	0.5	0.4	1.566

where we get a maximum value of f_{QoS} (product of number of users and throughput of the system). Since the objective of the proposed slot allocation algorithm is to operate the system at the point where f_{QoS} is maximum we need to have an analytical expression for k_{avg} and k'_{avg} as a function of

The results in Table 4.1 show that the value of k_{avg} and k'_{avg} remain constant even with a change in p_l. Since k_{avg} varies in the range 0.6 to 0.7 let k_{avg} be fixed at 0.65, likewise since k'_{avg} varies in the range of 0.5 to 0.6 let k'_{avg} be fixed at 0.55 (ignoring the data of $p_l = 0.1$) we use these values of k_{avg} and k'_{avg} for the remaining set of experiments. The performance of the new scheme is compared with the performance of IS-136 system which is the wideband TDMA based standard in United States [54], and a new scheme where all the slots in the frame are Mode-1 slots.

Simulation experiments are carried out for varying values of p_l but for each experiment the value of p_l is fixed. The value of F which is the tolerance limit of the system for the change in airlink frame loss rate was fixed to 0.05.

4.3.2 Simulation Results for Number of Users

Figures 4.5-4.7 present the number of users carried in each slot in an IS-136 system, a system using the new algorithm and a system having only Mode-1 slots. These results are obtained for varying airlink frame loss probability and with varying number of requests at the end of each frame.

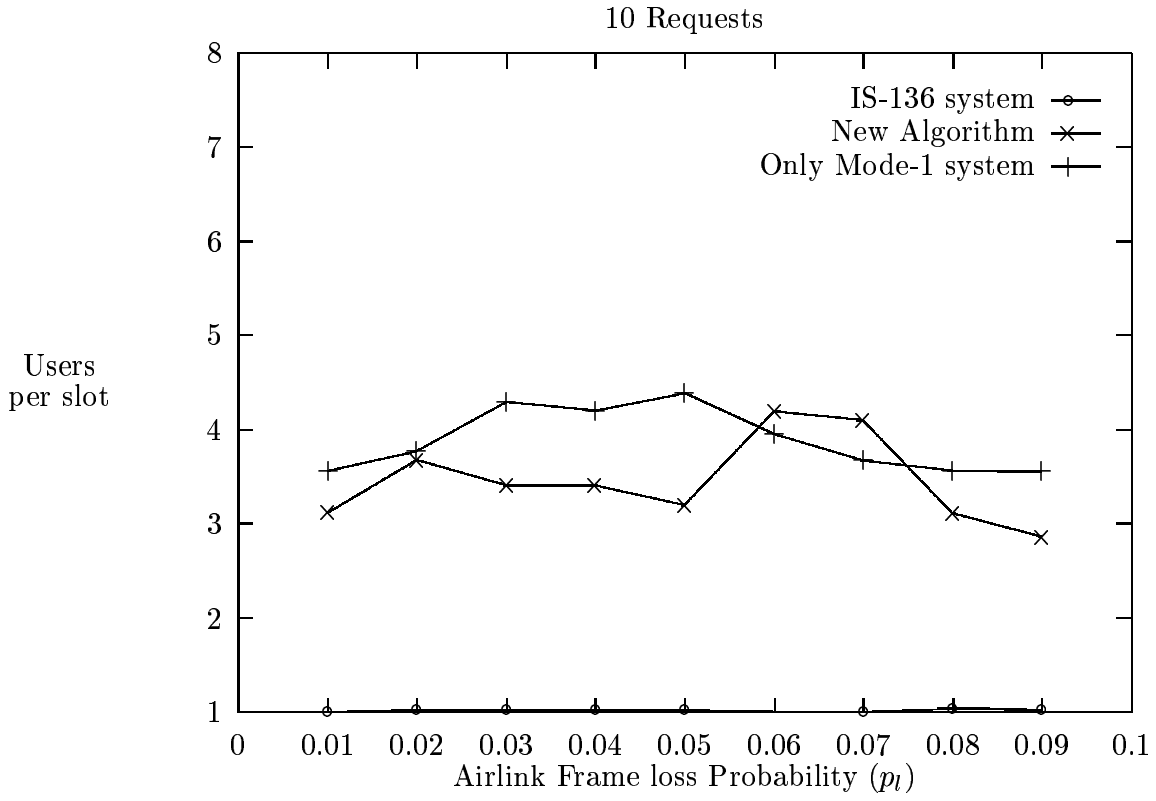

Figure 4.5: Number of Users per slot with varying airlink frame loss probability and 10 requests per frame

The number of users served in each slot remains constant for an IS-136 system and it does not change with varying airlink frame loss rate and with the number of

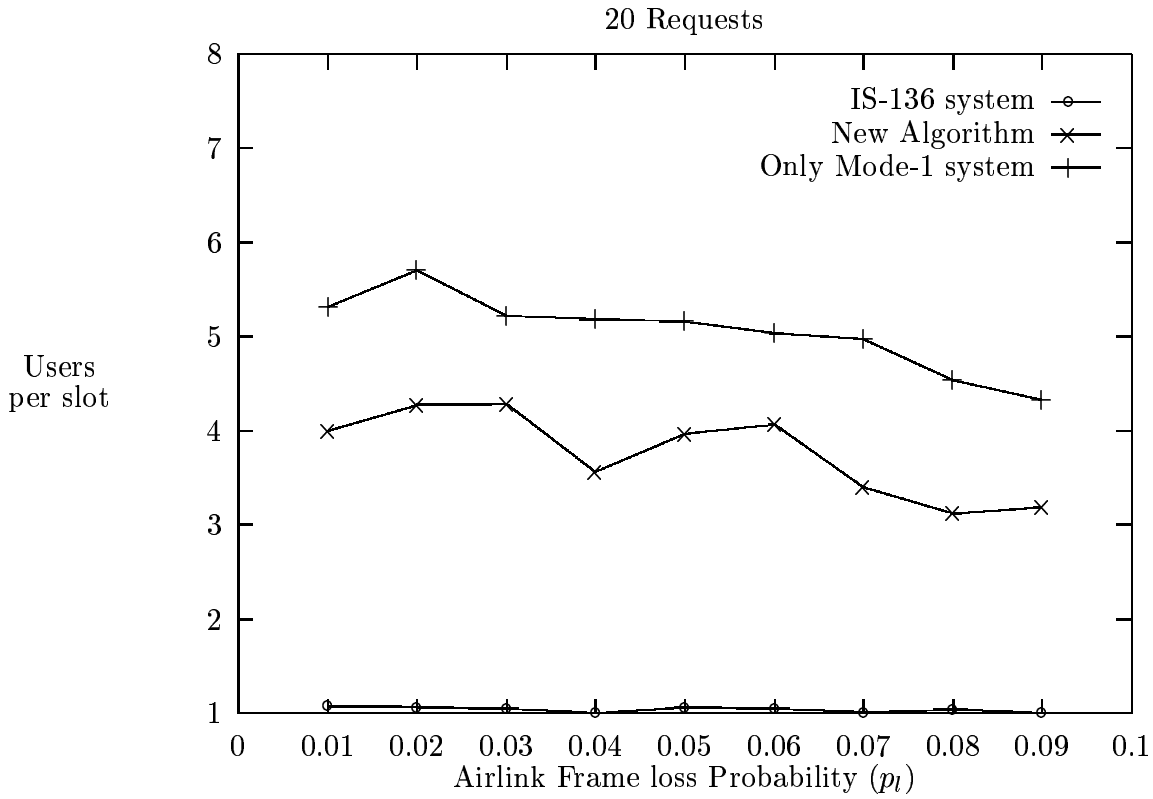

Figure 4.6: Number of Users per slot with varying airlink frame loss probability and 20 requests per frame

user requests at the end of each frame. In a system using the new algorithm and the system using only Mode-1 slots, with an increase in the number of user requests the number of users supported in a slot increases and the number of users supported in a slot goes down with an increase in airlink frame loss probability. In the case of IS-136 system the frame structure is fixed so with a change in p_l there is no change in the number of user slots. Hence, the number of users supported remains the same. In the case of the system using the new algorithm and the system with

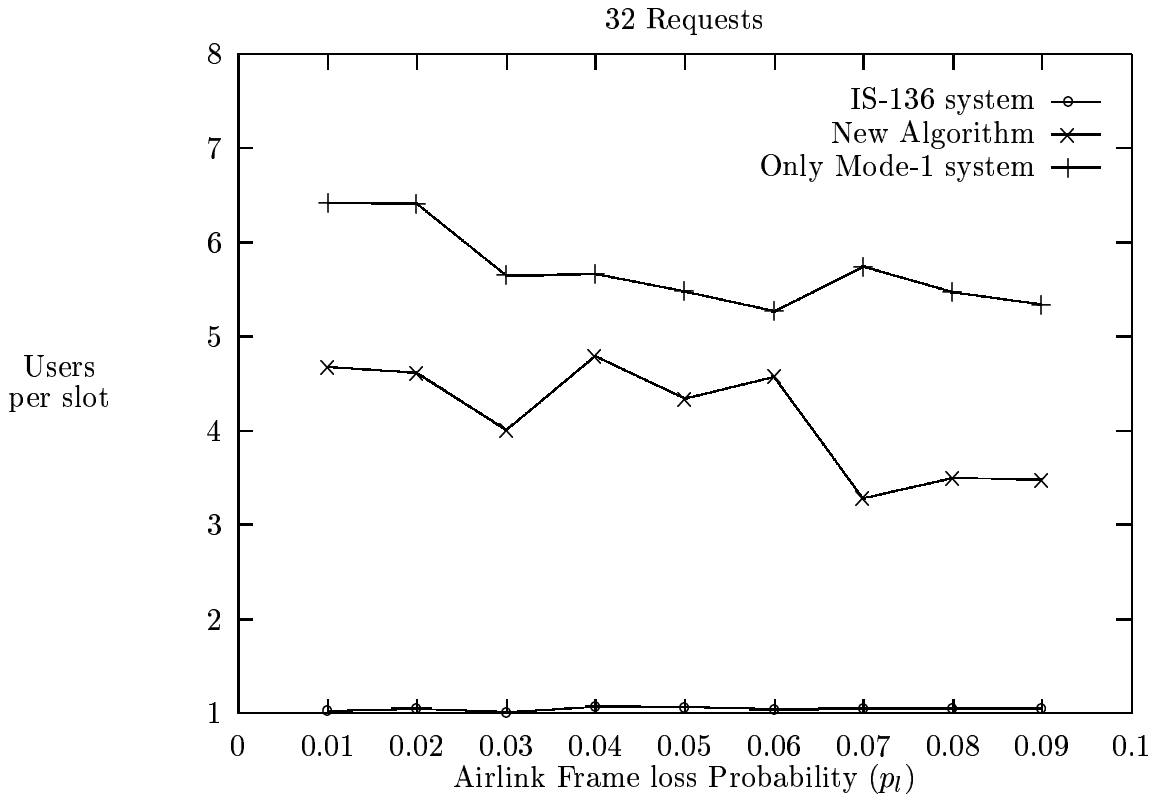

Figure 4.7: Number of Users per slot with varying airlink frame loss probability and 32 requests per frame

only Mode-1 slots there is a change in the number of minislots allocated to users. With an increase in airlink frame loss probability there is a decrease in the number user minislots per slot (Equation 3.3). Hence, the number of users supported in each slot goes down. Moreover, for the new algorithm system with an increase in the airlink frame loss probability the number of Mode-2 slots increases (Equation 3.13) to increase the throughput of the system. Hence, the number of users supported in each slot decreases. For an IS-136 system to assign user requests to the slots

the linear programming formulation for Mode-1 slots is used. The emphasis of this formulation is to minimize the bandwidth wasted. Hence, the system makes a choice of choosing the maximum bandwidth application, so the number of users supported per slot remains unchanged even with an increase in the number of requests at the end of each frame. In the system using the new algorithm and a system with only Mode-1 slots with an increase in the number of user requests there is a better choice of the combination of users to satisfy the bandwidth constraint and the number of users constraint. Hence, we find an improvement in the numbers of users served in each slot.

4.3.3 Simulation Results for Throughput of the system

Figures 4.8-4.10 plot the throughput results of IS-136 system, a system using the new algorithm and a system with only Mode-1 slots. These results are obtained for varying airlink frame loss probability and with varying number of requests at the end of each frame.

The throughput of IS-136 system goes down with an increase in the airlink frame loss probability. With a change in the number of requests at the end of each frame there is no change in the throughput in an IS-136 system. The throughput of the system using the new algorithm remains constant with a change in airlink frame loss probability and with a change in the number of user requests at the end of each frame. In the system using Mode-1 slots only the throughput remains constant with a change in the number of user requests at the end of each frame and it decreases with an increase in the airlink frame loss probability. IS-136 system has a fixed frame structure hence there is a drop in the throughput with increase in airlink frame loss probability. As shown in Figures 4.5-4.7 there is no change in the number of users

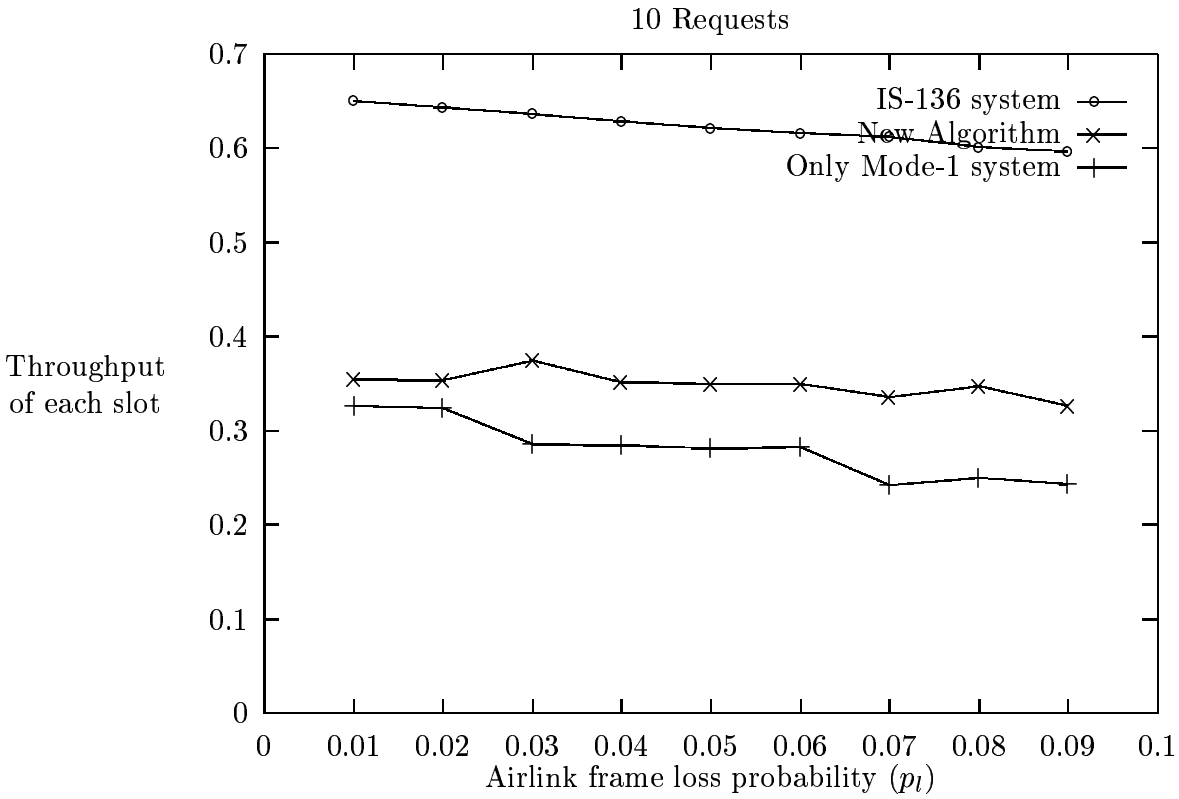

Figure 4.8: Throughput of the system with varying airlink frame loss probability and 10 requests per frame

supported with a change in the number of user requests available at the end of each frame because it always chooses a higher data rate application. Hence, there is no change in the throughput with a change in the number of user requests at the end of each frame. In the system using the new algorithm with a change in airlink frame loss probability the system adjusts the number of Mode-1 and Mode-2 slots and ensures that there is no change in the throughput of the system. This means that there will be a decrease in the number of users in each slot with a change in airlink frame

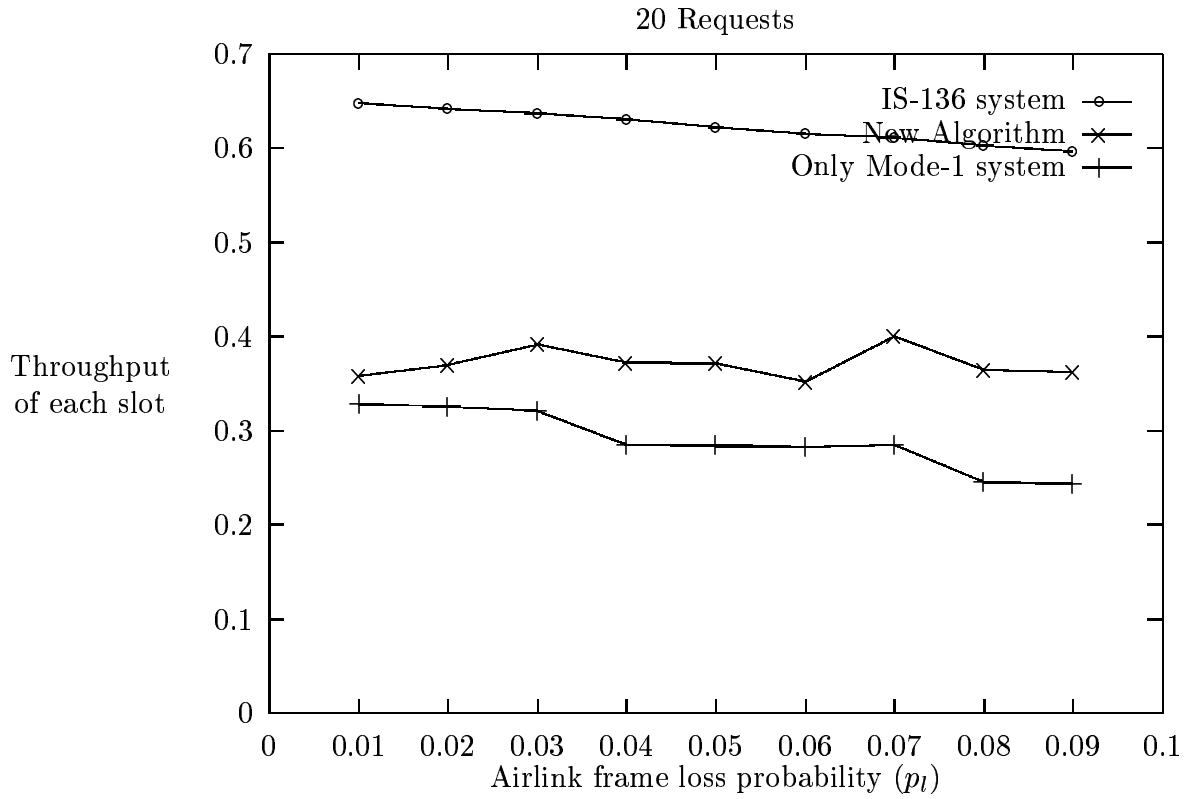

Figure 4.9: Throughput of the system with varying airlink frame loss probability and 20 requests per frame

loss probability. This is observed in the plots in Figures 4.5-4.7. Like wise with a change in the number of user requests at the end of the frame the system optimizes f_{QoS} function by finding a combination of users which would increase the number of users in the system without a change in the throughput of the system. In the system having only Mode-1 slots with an increase in airlink frame loss probability there is an increase in the number of retransmission minislots, which results in a decrease in the throughput of the system.

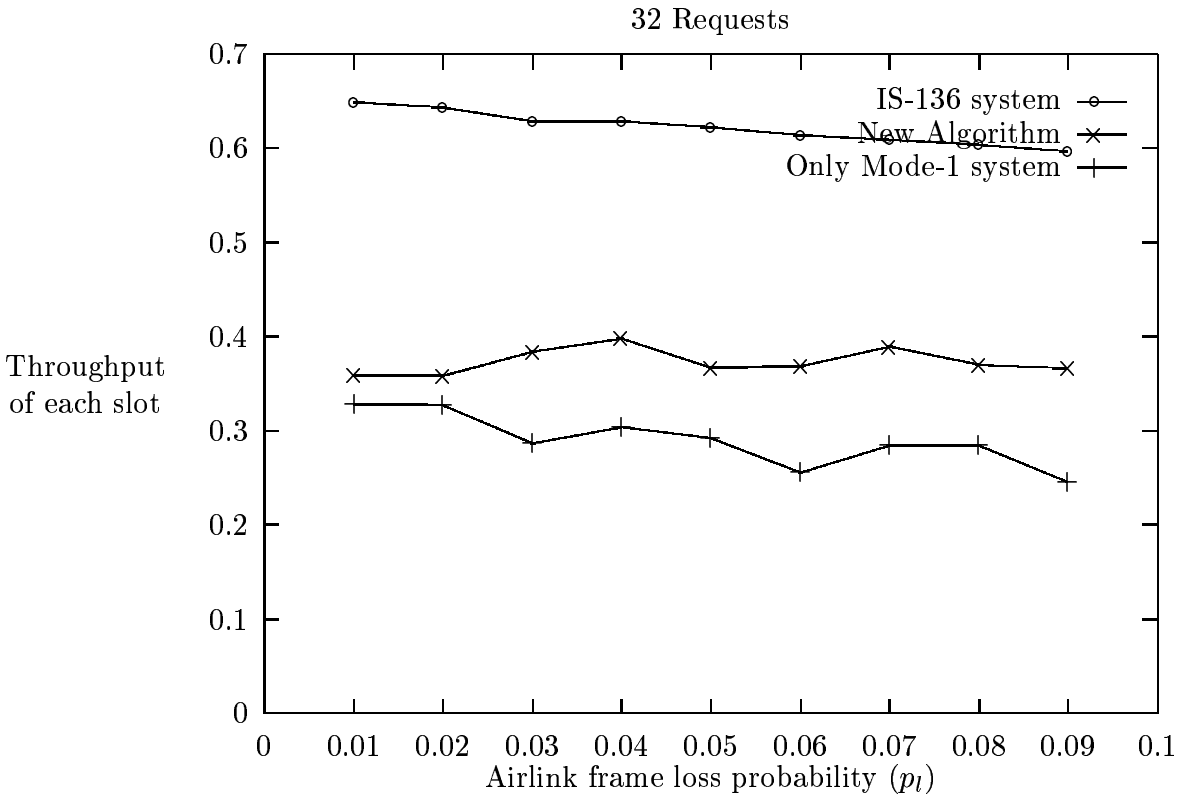

Figure 4.10: Throughput of the system with varying airlink frame loss probability and 32 requests per frame

4.3.4 Simulation Results for deviation in interpacket delay and Standard deviation in the deviation of the Interpacket Delay

The linear programming formulation for Mode-1 slots (Chapter 3) ensures that all the users are allowed to send data in each frame and they are allocated minislots as per their bandwidth requirements. The next important parameter affecting the quality of service for the system is the inter-packet delay. For the Mode-1 users, the number of retransmission minislots are used to transmit their lost packets. Each

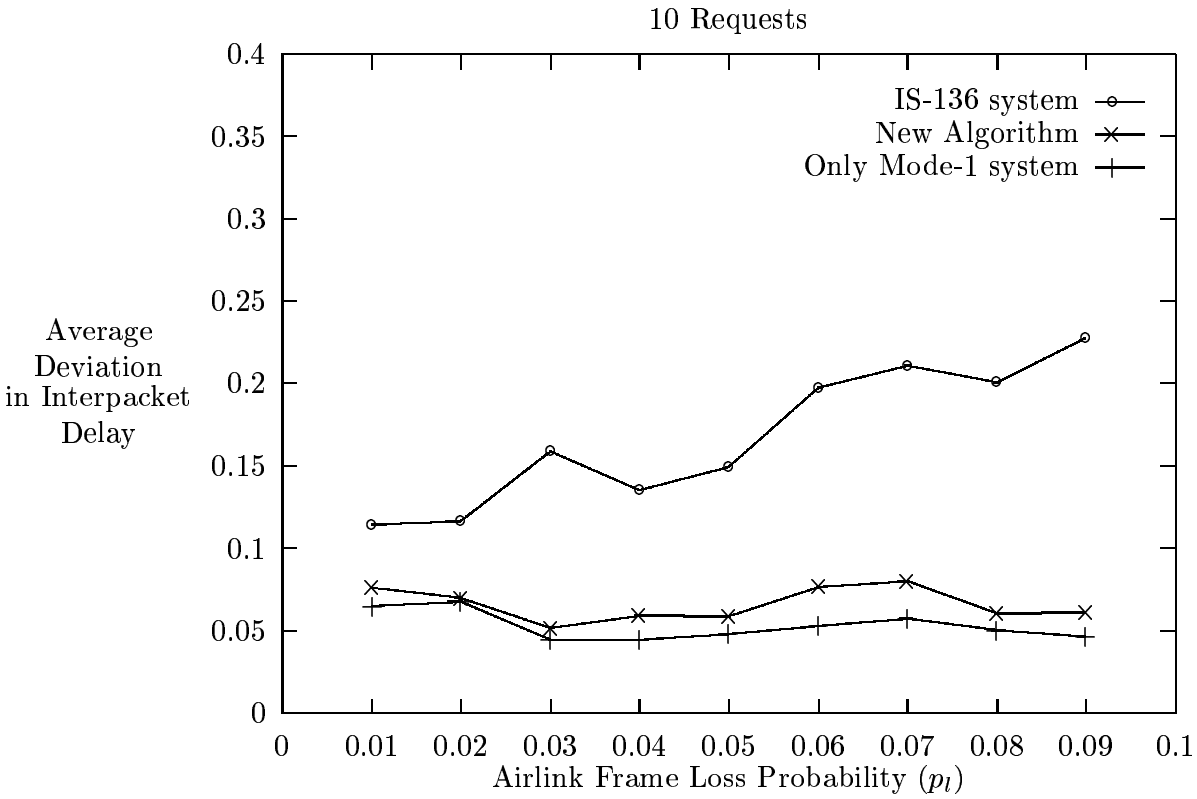

Figure 4.11: Deviation in interpacket delay with varying airlink frame loss probability and 10 requests per frame

slot has certain number of minislots such that on the average each user transmits his packet of information in a fixed number of retransmission attempts. To compute the interpacket delay assume a fixed packet size called Maximum Transmission Unit ($\mathcal{MTU}$). One way of choosing the size of $\mathcal{MTU}$ would be, fix the size of $\mathcal{MTU}$ as the size of the maximum size of a TCP segment. For each Mode-1 user compute the average number of frames required to transmit $\mathcal{MTU}$ number of data bits. Plot the average and standard deviation of the difference between the average delay and the

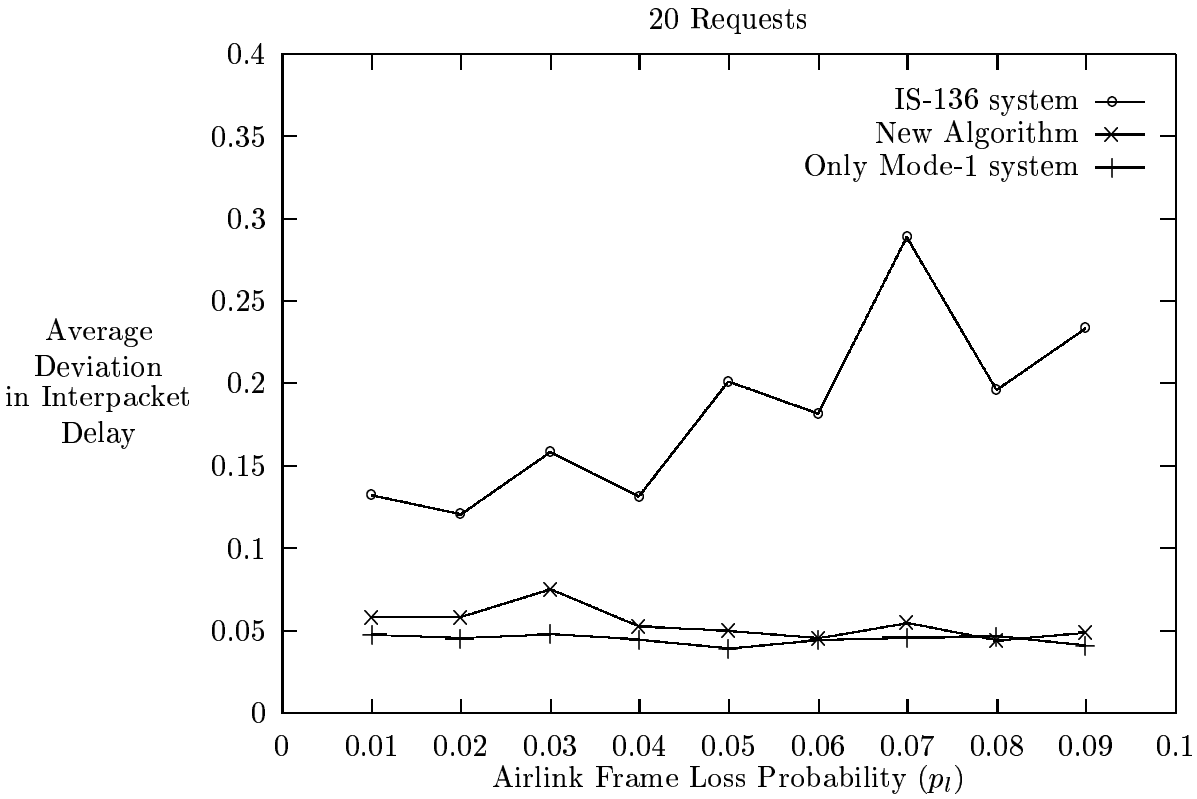

Figure 4.12: Deviation in interpacket delay with varying airlink frame loss probability and 20 requests per frame

delay which is the number of frames required to send an $\mathcal{MTU}$ data unit when the airlink frame loss probability is zero. For the results presented $\mathcal{MTU}$ is set to 1944 bits which is the amount of data that can be sent in 40 msec which is the time taken to send two TDMA blocks (IS-136 system [55]) over a 48.6 Kbps channel [55].

Figures 4.11-4.16 plot the results of the normalized average deviation and standard deviation in the normalized average deviation of the delay experienced in an IS-136 system, a system using the new algorithm and a system with only Mode-1 slots

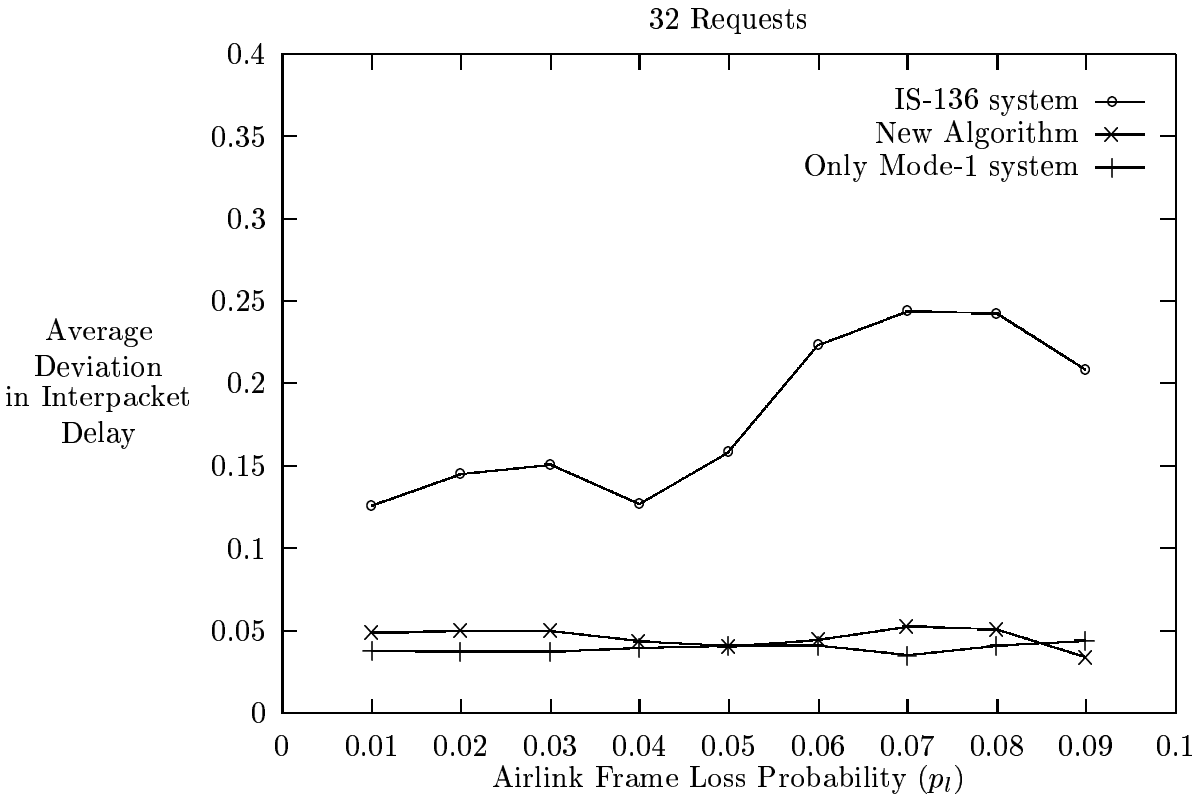

Figure 4.13: Deviation in interpacket delay with varying airlink frame loss probability and 32 requests per frame

These results are obtained for varying airlink frame loss probability and with varying number of requests at the end of each frame. In an IS-136 system since there are no retransmission slots with an increase in the airlink frame loss probability the time taken to send a packet increases. Hence, the average deviation of delay increases with an increase in airlink frame loss probability. With an increase in airlink frame loss probability there is wide fluctuation in the variation of the delay. Hence, the standard deviation of the deviation in the delay to send $\mathcal{MTU}$ amount of data increases. For

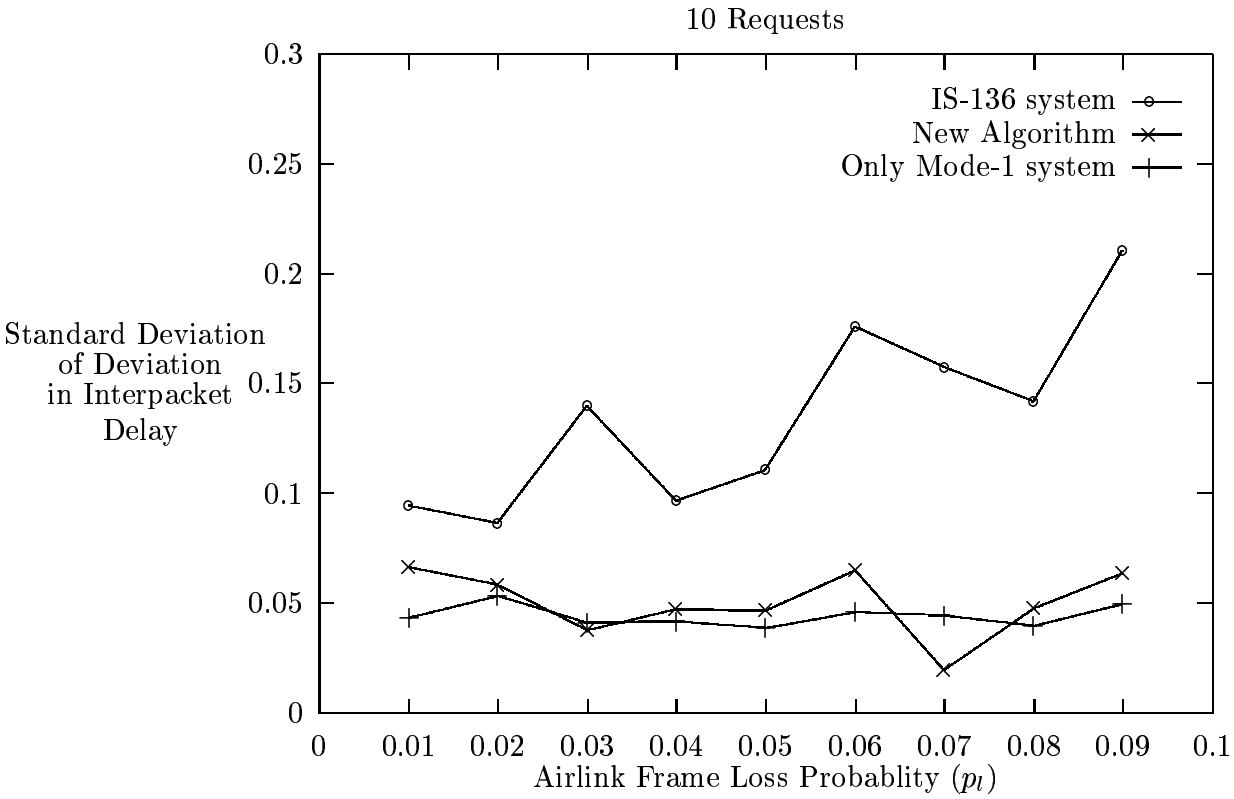

Figure 4.14: Standard Deviation of deviation in interpacket delay with varying airlink frame loss probability and 10 requests per frame

the system using the new algorithm or the system using only Mode-1 slots there is no change in the delay or variation in the delay with a change in airlink frame loss probability because of the retransmission minislots. The number of minislots in each slot are designed to compensate the increase in delay due to an increase in the airlink frame loss probability.

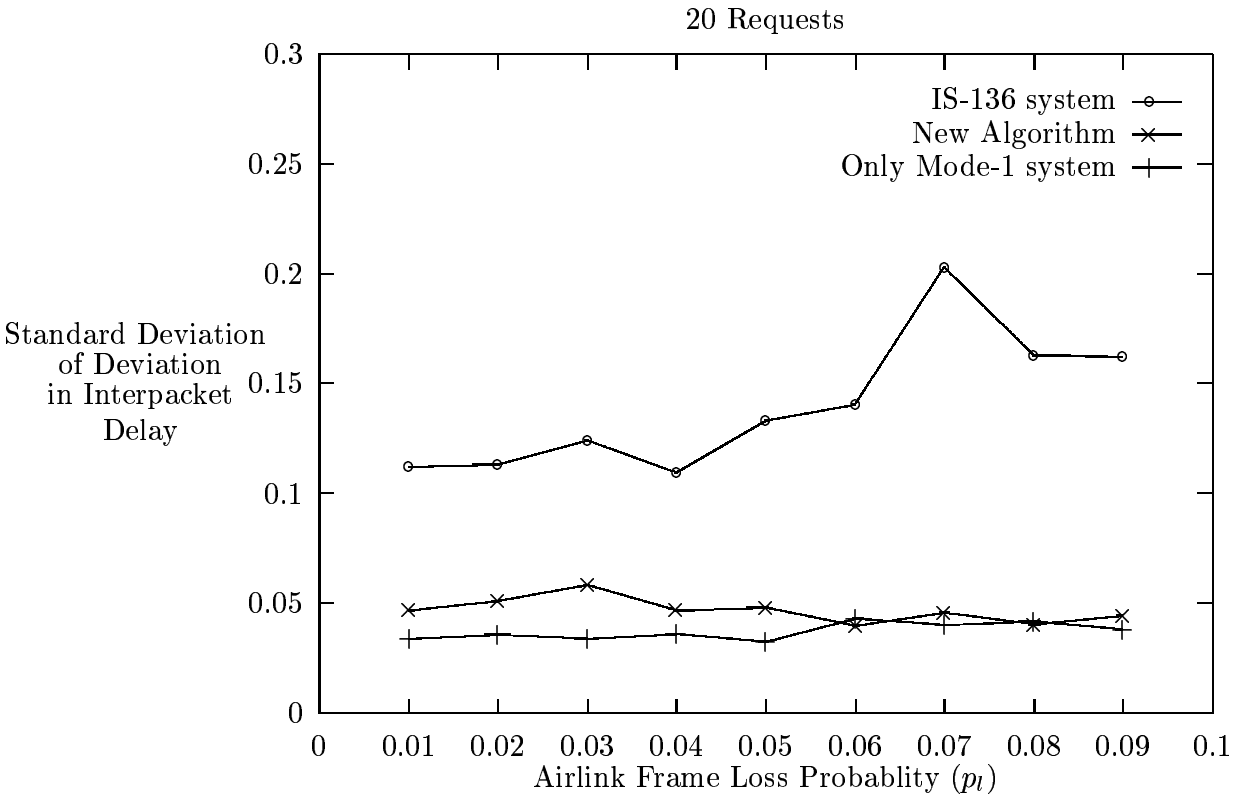

Figure 4.15: Standard Deviation of deviation in interpacket delay with varying airlink frame loss probability and 20 requests per frame

4.4 Dynamic Characteristics of the System

The results presented in Figures 4.5-4.16 show the performance of IS-136 system, system using the new algorithm, system using only Mode-1 slots for varying airlink frame loss probability and for varying number of user requests at the end of each frame. Each value in these plots is obtained by fixing the airlink frame loss probability for that simulation experiment. In real networks that is never the case. Given the link characteristics of wireless links the airlink frame loss probability fluctuates over

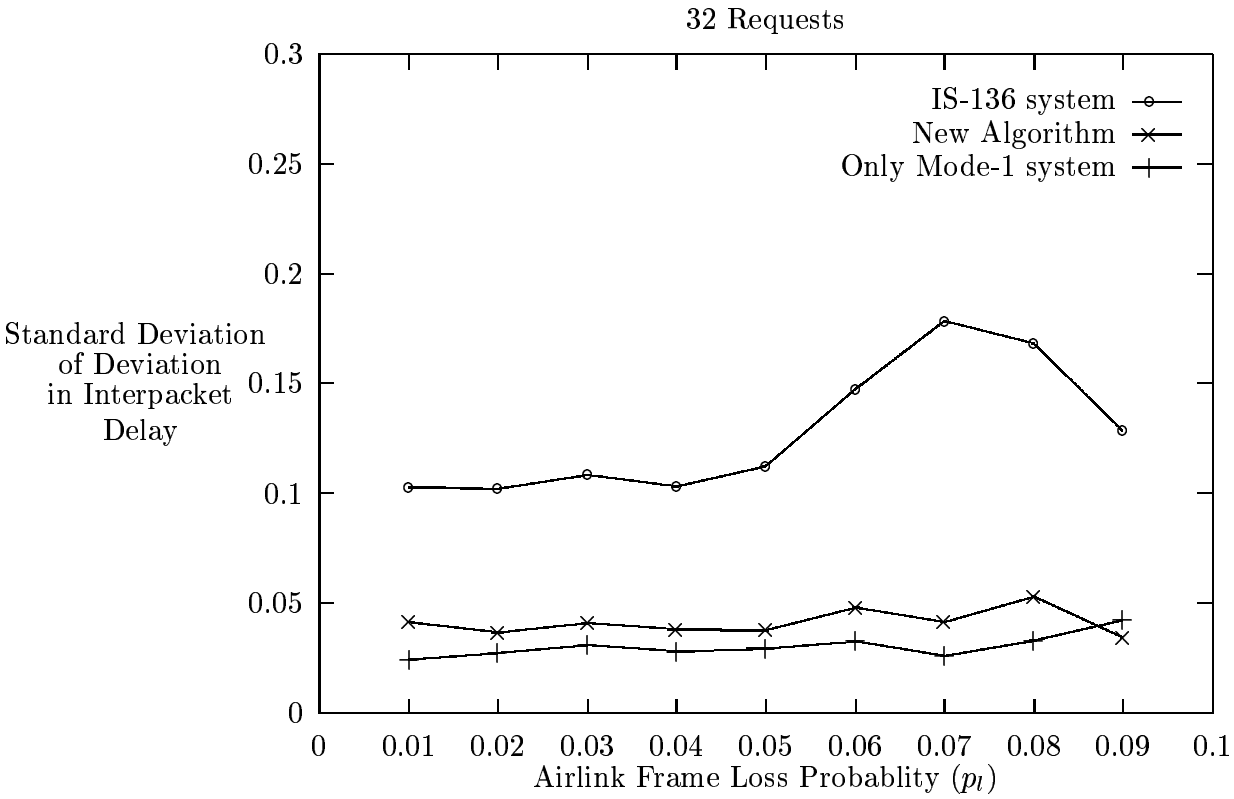

Figure 4.16: Standard Deviation of deviation in interpacket delay with varying airlink frame loss probability and 30 requests per frame

the range (0 to 0.15) with a mean around 0.03. To evalute the performance of the system under these conditions we conduct experiments in which the airlink frame loss probability (p_l) is changed after certain number of simulation cycles. Due to the change in p_l the number of users in each slot is not going to change immediately because if a user is already there in the system he is not going to be dropped and if a user is assigned certain amount of bandwidth a portion of his bandwidth cannot be assigned to another user until he terminates his call. We cannot observe an immediate

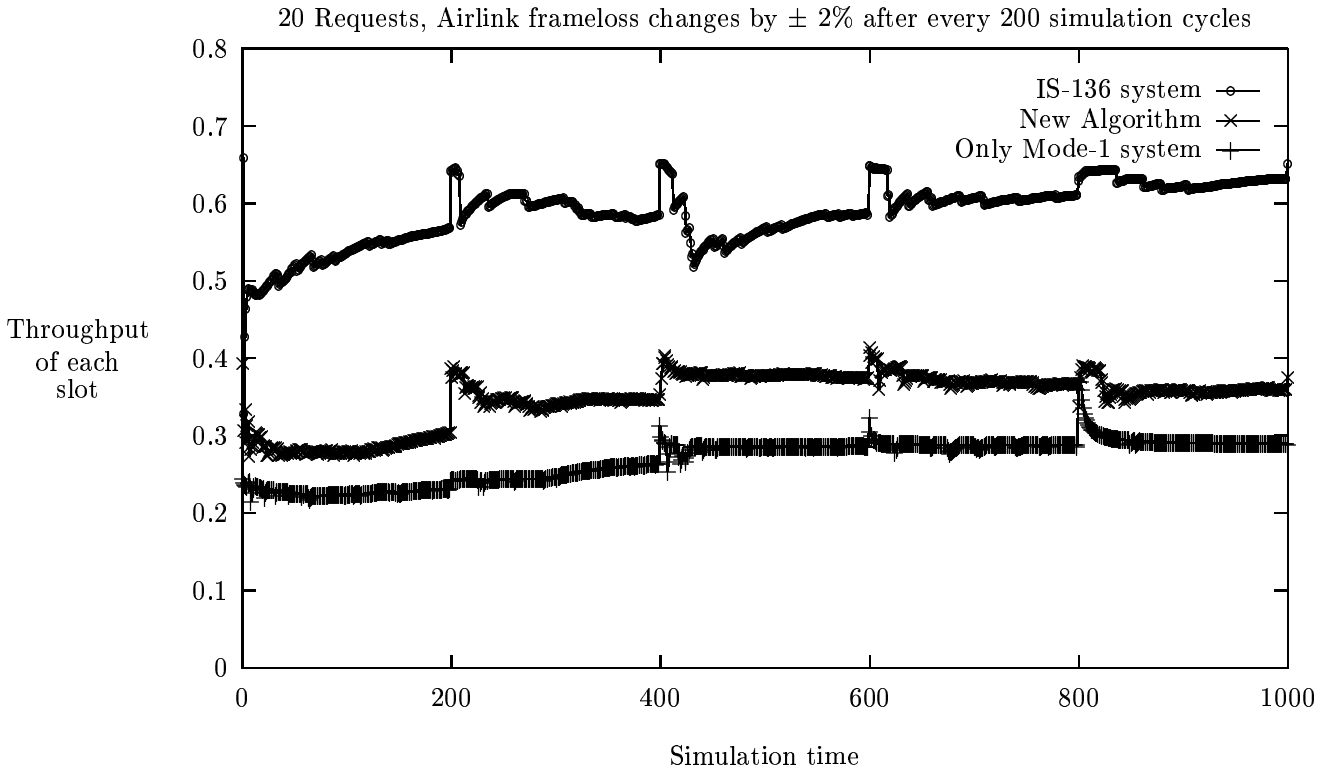

Figure 4.17: Throughput of the system at each simulation cycle at the start of the simulation

change in the interpacket delay with a change in p_l because the interpacket delay is computed only after an $\mathcal{MTU}$ amount of data is sent. Throughput of the system is computed at the end of each simulation cycle. With a change in p_l looking at the variation of the throughput of the system one can study the dynamics of the system. For the results presented in Figures 4.17-4.20 the airlink frame loss probability is changed by ± 0.02 and p_l can have a maximum value of 0.2. The airlink frame loss probability is either increased or decreased after certain number of simulation cycles with equal probability.

Figure 4.17 plots the throughput of IS-136 system, system using the new algorithm

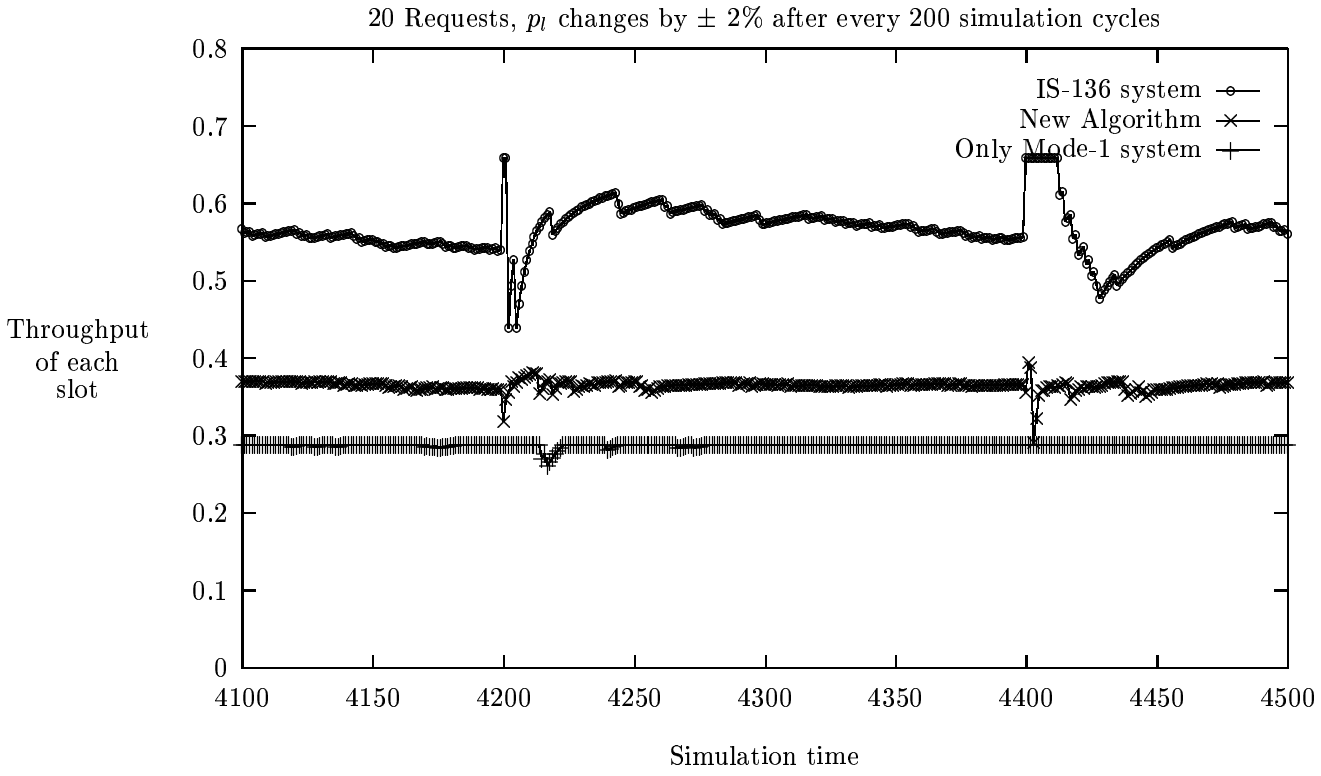

Figure 4.18: Throughput of the system at each simulation cycle once the simulation is stabilized

and system using only Mode-1 slots at the end of each simulation cycle at the start of the experiment. In this experiment p_l is changed after every 200 simulation cycles. The throughput of the system is not stabilized and with a change in p_l the system slows down in attaining its steady state throughput. However, the IS-136 system takes more simulation cycles to get back to its steady progress to attain its steady state throughput. For the system using the new algorithm or using only Mode-1 slots because of the retransmission minislots which are adjusted dynamically with a change in p_l the throughput does not get effected for a long time, i.e., the system stabilizes quickly.

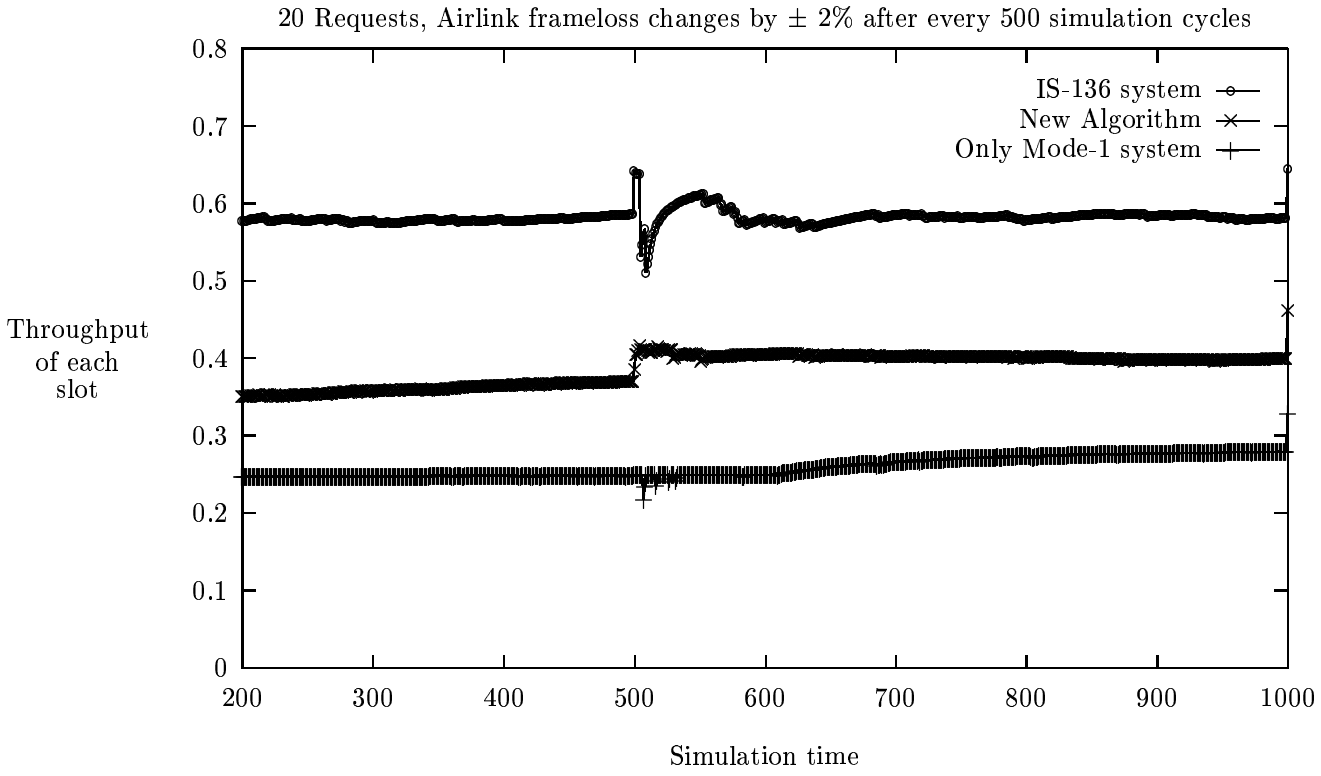

Figure 4.19: Throughput of the system at each simulation cycle at the start of the simulation

Figure 4.18 plots the throughput of the systems after they have reached a steady state. IS-136 system takes more simulation cycles for it to adjust to a change in p_l and return to its steady state throughput. System using the new algorithm or using only Mode-1 slots adjust to the change in p_l in few simulation cycles. This is because in a system using the new algorithm and the system using only Mode-1 slots the number of retransmission minislots in each Mode-1 slot is changed dynamically. Moreover, the number of Mode-1 and Mode-2 slots in the frame are changed dynamically in a system using the new algorithm.

Figure 4.19 plots the throughput of the system at the end of each simulation

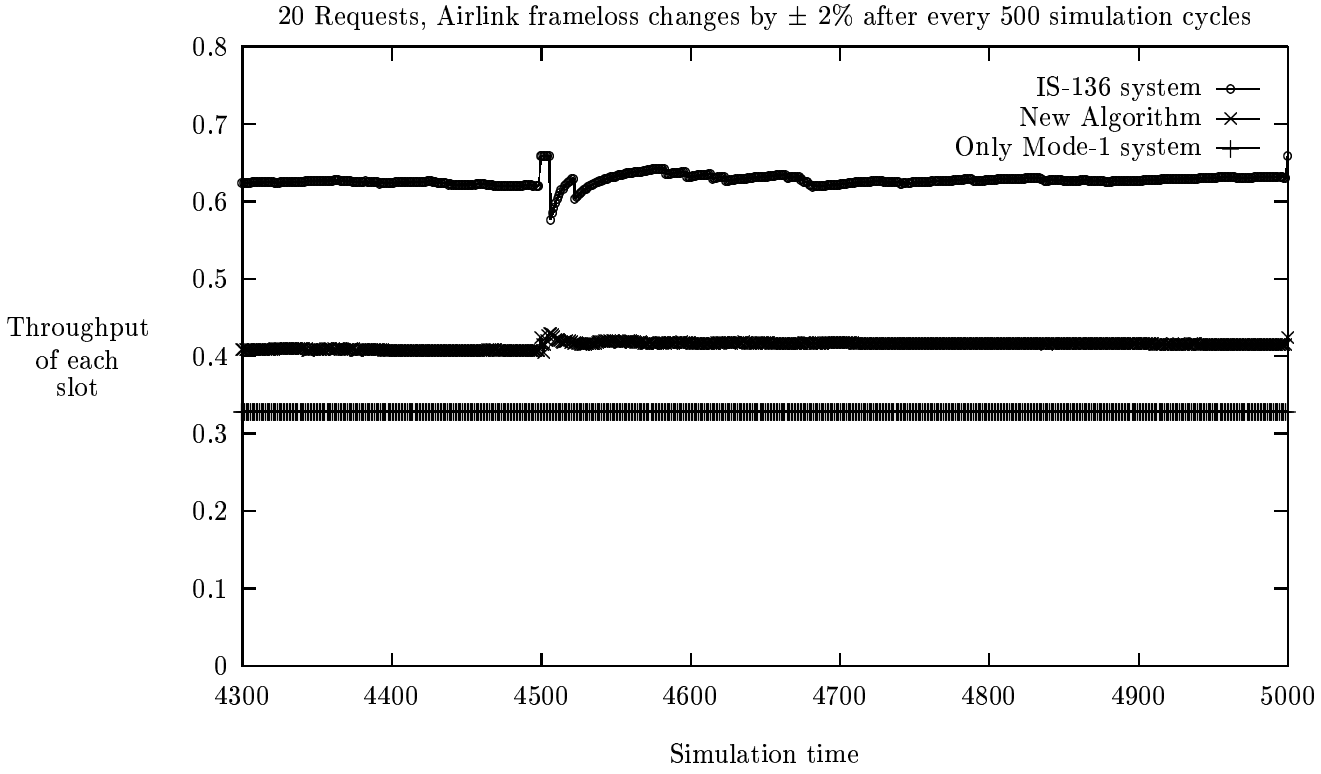

Figure 4.20: Throughput of the system at each simulation cycle once the simulation is stabilized

cycle at the start of the experiment. In this experiment p_l is changed after every 500 simulation cycles. The IS-136 system takes more simulation cycles to adjust to the change in p_l. The system using the new algorithm and the system using only Mode-1 slots adjust to the change in few simulation cycles. Comparing the results in Figure 4.17 where p_l is changed after every 200 simulation cycles with the results in Figure 4.19 where p_l is changed after every 500 simulation cycles, IS-136 system takes more simulation cycles to reach a steady state does not reach a steady state before the next change in p_l occurs. In wireless networks change in p_l occur very often, so if IS-136 were used in that environment the system would be very unstable. Figure 4.19

plots the throughput at the end of each simulation cycle after the system has reached a steady state. With a change in p_l IS-136 system takes more simulation cycles to attain its steady state value than the systems using the new algorithm or the system using only Mode-1 slots.

4.5 Summary

In this chapter we presented the dynamic slot reallocation algorithm, the simulation setup and the results of the simulation experiments. Results show a considerable improvement in the system performance *w.r.t* the number of users carried by the system and the reliability in meeting the QoS requirements. IS-136 performs better than the new algorithm *w.r.t* the system throughput. Inspite of higher throughout of the system, the system cannot meet the inter packet delay requirements of user requests. Moreover, for a fluctuation in the airlink frameloss probability IS-136 system performs poorly in the time taken to reach a steady state. The new algorithm adpats the frame structure and adjusts the user requests in the system when ever there the air link frameloss probability changes by a fraction $\pm$ F.

CHAPTER 5

QOS BASED CALL ADMISSION CONTROL ALGORITHM

5.1 Introduction

This chapter presents a Call Admission Control (CAC) algorithm for next generation wireless networks. The proposed CAC algorithm is not based upon a priori knowledge of the source characteristics, and admits a call ensuring that the service commitments of the ongoing calls are not violated. The CAC algorithm consists of two parts. The first part adapts the service rate of the system based upon the number of frames lost during the observation period. The second part deals with estimating expected delay for each class of service upon admitting a new call.

Many designs for integrated service networks offer a bounded delay packet delivery service to support real-time applications [8, 9, 10, 11, 28, 12, 13, 14, 15, 16]. To provide bounded delay service, networks must use admission control algorithm to regulate their load. Previous work [26, 27] on admission control mainly focused on algorithms that compute the worst case theoretical queuing delay to guarantee an absolute delay bound for all packets. This chapter describes a statistical measurement-based admission control algorithm which allows occasional delay violations.

Traditional real-time applications provide a hard or absolute bound on the delay of each packet. Typically these are referred to as guaranteed service applications and existing solutions provide some form of bounded packet delay service. Earlier work [34, 8, 9, 10, 11, 28, 12, 13, 14, 15, 16] shows that the ability of bounded delay services to achieve higher utilization and also to meet their service commitments depends crucially on the admission control algorithm. Conversely, the ability of an

admission control algorithm to increase network utilization is ultimately constrained by service commitments the network makes.

Most of the existing call admission control algorithms are designed to ensure QoS in wireline networks. Given the link characteristics of wireless networks, and call requests with different QoS specifications there is a need for a novel admission control algorithm which takes into consideration different classes of service having different delay constraints by adapting the system decision process to admit a call or not based upon the previous decisions and the performance obtained.

5.2 Previous Work

The imminence of new services with a broad range of burstiness characteristics and their integration through statistical multiplexing have focused on call admission schemes as the prime instrument for rate-based congestion control. By preventing admission of an excessive number of calls or sources to the multiplexer, call admission policies strive for a balance between quality of service (QoS) and efficient use of network resources. Designs based on peak rates and mean rates result in two extremes of the network performance. As a compromise, Elwalid and Mitra [34] proposed the *effective bandwidth* approach in which a user demand is characterized by the effective bandwidth rather than the peak or the mean rate which bound the performance of the effective bandwidth approach. Many real-time applications such as *vat, nv,* and *vic* [38], have been developed for packet-switched networks. These applications adapt to actual packet delays and are thus tolerant to occasional delay bound violations in the sense that they do not need an absolute reliable bound. It is important to note that the service definition itself does not specify the acceptable level of delay violations because reliably ensuring that the failure rate does not exceed a particular

level leads to worst-case calculations.

Traditional approaches to admission control, like those used for guaranteed service, use a priori characterizations of sources to calculate the worst-case behavior of all the existing flows in addition to the incoming one. Calculating the worst-case delays is very complex as the worst case scenario Is not defined. Inspite of this the underlying admission control principle is conceptually simple, it determines whether granting a new request for service leads to a violation of any delay bound ? [8]. Network utilization under this model is usually acceptable when flows are smooth (mean rate is same as peak rate). However, when flows are bursty, their traffic characterizations must necessarily be quite loose, in that the average behavior of the flows is significantly less than the upper bound of the traffic descriptions and guaranteed service inevitably results in low utilization [17].

There are many other approaches to admission control that attempt to achieve higher utilization by weakening the degree of reliability of the delay bound. For instance, the probabilistic delay bound service described in [18] does not provide for the worst-case scenario, instead it guarantees a bound on the probability of lost/late packets based on statistical characterization of traffic [19]. In this approach, each flow is allotted an effective bandwidth that is larger than its average rate but less than its peak rate, thus increasing the network utilization. In most cases the a priori characterization of flows is based on a statistical model [20], [21] or on a fluid flow approximation [22], [23]. But it is almost impossible to provide accurate and tight statistical models for each individual flow. Therefore, the *a priori* traffic characterizations handed to admission control will be fairly loose upper bounds.

We believe that measurement based admission control will play a key role in

achieving high network utilization. The measurement based admission control approach advocated in [24], [25] uses an a priori source characterizations only for incoming flows and uses measurements to characterize those flows that are in the system. By making a decision based upon the statistical measurements of the source rather than its a priori source characteristics one can reduce the error in the description of the source characteristics which results in better network utilization and ensures that the source does not have to face QoS violations even if the source description is not accurate. Since the call admission depends upon the measurements and as the source behavior is not always static, the measurement based approach to admission control can never provide the complete reliable delay bounds needed for guaranteed (or even probabilistic) service. The system uses the measured data to compute the performance of the system using a queuing model, compare the performance obtained from the queuing model and the actual performance which is measured at the end of each frame to predict the performance of the system. In this way one can adapt the system to the source characteristics. This is the essence of the call admission control algorithm to be described in this chapter. However, note that the queuing model used to estimate the performance of the system is not based upon the a priori distribution of the sources.

All the admission control algorithms are evaluated under a scheduling discipline. In order to provide QoS guarantees for real-time traffic, several service scheduling disciplines have been proposed in the literature: Delay Earliest-Due-Date [8], Jitter Earliest-Due-Date [9], Rate-Controlled Static Priority Queuing [10], Virtual Clock [11], Packet-by-Packet Generalized Processor Sharing [28], [12], Stop-and-Go [13], [14], Hierarchical Round Robin [15], and Leave-in-Time [16]. The proposed call admission control algorithm will be evaluated under Higher Priority Class Packets First

scheduling scheme. The details of how priority is assigned to different classes of traffic are described in later sections.

5.3 Measurement Based Admission Control for Wireless Links

The admission control algorithm presented in [26] consists of two logically distinct aspects. The first aspect is the set of admission control criteria for when to admit a new flow. These criteria are based on an approximate model of traffic flows and use measured quantities as inputs. The second aspect is the measurement process itself.

To achieve a reliable bound that is less conservative, an approximation is made in determining the maximal delay of flows by replacing the worst-case parameters with measured quantities. This approximation is termed as *equivalent token bucket filter*. This approximation yields a series of expressions for the expected maximal delays that would result from the admission of a new flow. When admitting a flow, the admission control algorithm decides whether that flow can get requested service, and determines if by admitting the flow will there be any violation of prior network commitments.

To compute the effect of a new flow on the existing traffic, the algorithm presented in [26] models for the worst-case delay of priority queues. The formulation developed in [27] provides a tight bound on the worst-case delay for each of the priority queues. But [26] used the formulation which was proposed in [28] as it is simpler, but it is a looser bound for the worst-case delay. As proposed in [26] a call is not admitted if either one of the following conditions occur:

C1: The sum of the flows requested and current usage exceed the link capacity.

C2: On admission there is a violation of delay bounds of admitted calls.

Moreover the algorithm uses a token bucket traffic shaper to control the peak rate of packet transmission.

The drawbacks of this algorithm are:

- The wireless link is prone to higher rate of packet losses than a wire- line network. In the algorithm presented does not compensate this wireless property when making a decision whether to admit the call or not.
- Although the system updates the delay for each class of traffic by monitoring the queue, the formulation used to compute the delay has to adapt based upon the error in the obtained statistical values and the computed value using a traffic model for the source.

5.3.1 Our New Algorithm (NA)

To overcome the drawbacks in [26] we propose two algorithms:

$\mathcal{A}_1$: Adapt the service rate of the system based upon the number of frames lost during an observation period of N frames.

$\mathcal{A}_2$: In estimating the delay while admitting a new call, rather than basing the estimate completely on the queuing model assumed, it is better to adapt the system estimate based upon the earlier estimate and the observed performance.

The notations used:

N: Number of frames after which service rate of the system is updated.

μ: Maximum service rate of the system.

Δ_μ: Fraction by which the service rate may be altered after N frames, and it is equal to $\frac{\mu}{N}$.

μ_{avg}: Current service rate of the system.

E_N: Number of frames in error in the last set of N frames.

Service rate adaptation methodology

In this section, we present the service rate adaptation algorithm. In a set of N consecutive frames the system computes the number of frames lost. If the number of frames lost is larger than the fraction $\frac{\mu - \mu_{avg}}{\Delta_\mu}$ i.e., the system has dropped more number of frames than it should have dropped as per its service rate, Lower the service rate of the system by the quantum Δ_μ. If the number of frames dropped is less than the fraction i.e., the system dropped fewer number of frames than what it should have as per the current service rate, increment the service rate of the system by the quantum Δ_μ. However, if the service rate of the system is not greater than Δ_μ then the system does not decrement the service rate even if the number of frames dropped is larger than the fraction $\frac{\mu - \mu_{avg}}{\Delta_\mu}$. Otherwise this would mean the service rate of the system is less than or equal to zero. Run this algorithm after a set of N frames.

Algorithm for Service Rate Adaptation

$\mu_{avg} = \mu$

After a set of N frames

if $(\mu_{avg} > \Delta_\mu)$ **then**

if $(\frac{\mu - \mu_{avg}}{\Delta_\mu} < E_N)$ **then** (The last set of N frames has more number of frames in error than the estimate made at the beginning of the set)

$\mu_{avg} = \mu_{avg} - \Delta_\mu$ (Reduce the service rate
by a quantum)

else

if ($\frac{\mu - \mu_{avg}}{\Delta_\mu} > E_N$) **then** (The last set of N frames has less
number of frames in error than the estimate made at
the beginning of the set)

$\mu_{avg} = \mu_{avg} + \Delta_\mu$ (Increment the service
rate by a quantum)

else

$\mu_{avg} = \mu_{avg}$ (If the number of frames in error in the last set
of N frames is same as the estimate made before the
set then do not change the service rate)

Note that the service rate adaptation algorithm does not reduce the service rate drastically upon discovering frame losses or increase it drastically when there are fewer frame losses. Instead the change is by a quantum Δ_μ and this is done only when the number of frames in error are more than the threshold which is set based on the current service rate (μ_{avg}). This enables the system to adapt to the network conditions gradually. In the scenario where the current service rate is modified by a large fraction, there is a wide fluctuation in the service rate this can result in more number of calls being dropped or more calls could be admitted which leads to frequent QoS violations.

Adaptive Delay Computation

Before admitting the call into the system the system computes the delay which will be experienced by the on-going calls if the new call is admitted. The authors of [26] used the model proposed in [28] to compute the effect on the delay on different classes of services when the new call is admitted. The problem in this model is, the model does not adapt to the traffic load and it is based upon a prior characterization of the source. In reality it is very difficult to characterize a source by a particular distribution.

Moreover the authors of [26] conclude their paper emphaisisng the need to have a methodology to compute the delay and this computation should be adaptable based upon the traffic load to get a better estimate. In summary the estimates obtained from a priori characterizations of the source are not accurate. When a new call arrives using an M/M/1 model the delay that each class of service has to experience for the cases when the call is admitted and when the call is not admitted is computed. The system uses the computed delay values using the current arrival and service rate, delay value using the arrival rate assuming that the new call is admitted and the current service rate and delay value using the arrival rate and the service rate when a call arrived the last time to fit into a quadratic function (3 points). The system then tries to estimate the expected delay if the new call is admitted by using a quadratic equation having the same second derivative as the earlier quadratic equation and using the current statistical delay and the statistical delay when a call arrived the last time. Note that we assume M/M/1 model just to get the trend in the variation of the delay.

Figure 5.1 shows the system model which is used for the simulation experiments. In this figure, $C_1, \cdots, C_n$ denote different classes of traffic with class C_i having

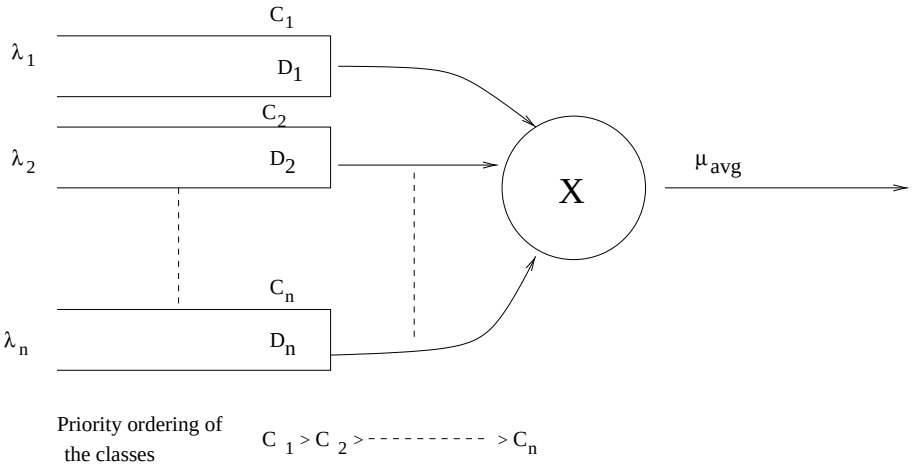

Figure 5.1: Assumed System Model

higher priority than class C_j if $i < j$. Let D_i^{class} be the tolerable maximum delay of the packets of class C_i service request, λ_i be the statistical arrival rate of packets, μ_i be the statistical service rate of packets and D_i be the statistical delay in the class C_i queue and $\hat{D}_i$ is the estimate of the statistical delay if the new call is admitted. Let D_p^i be the computed delay using M/M/1 model at the time of the arrival of the last call and $\hat{D}_p^i$ be the statistical delay at the time when the last call arrived, respectively. Packets are scheduled based upon their priority and within a class they are serviced using FIFO service model.

Assume a call request of class C_i be made and the call has a peak rate λ_i^p. The delay computation algorithm is described below:

Delay Computation Algorithm

for $j = 1$ to i **do**

$\hat{D}_j = D_j$ These classes are not effected due to the new call

$D_q^i = \frac{\lambda_i}{\mu_i - \lambda_i}$ Delay estimate without admitting the
new call using M/M/1 model

$D_{q_{new}}^i = \frac{\lambda_i + \lambda_i^p}{\mu_i - \lambda_i^p - \lambda_i}$ Delay estimate after admitting the
new call using M/M/1 model

$\hat{D}_i = \mathrm{Q}(D_i, \hat{D}_p^i, D_q^i, D_{q_{new}}^i, D_p^i)$

Expected delay computed using the second order derivative of the delay estimates using M/M/1 model and the two statistical delay values available.

for $j = i+1$ to n **do**

$D_q^j = \frac{\lambda_j}{\mu_j - \lambda_j}$ Delay estimate without admitting the
new call using M/M/1 model

$D_{q_{new}}^j = \frac{\lambda_j}{\mu_j - \lambda_i^p - \lambda_j}$ Delay estimate on admitting the
new call using M/M/1 model

$\hat{D}_j = \mathrm{Q}(D_j, \hat{D}_p^j, D_q^j, D_{q_{new}}^j, D_p^j)$

Expected delay computed using the second order derivative of the delay estimates using M/M/1 model and the two statistical delay values available.

For the case where there is no previous estimate for the delay the system does a linear fit of the delays obtined from M/M/1 model. To compute the expected stastical delay, the system uses the fact that the slope of the linear fit and the current statistical delay to estimate the expected statistical delay if the new call is admitted.

Call Admission Conditions

Every call is categorized into one of the n classes of service based upon the delay requirements of the service request. Since each class is associated with a certain delay guarantee, on arrival of a call the delay for packets in each class of service is

computed based upon the above formulation. The computed expected delay values for different classes are denoted as $\hat{D}_j$, for $1 \leq j \leq n$. For a call to be admitted the following two conditions are to be satisfied.

A_1: $for\ 1 \leq j \leq n$

$\hat{D}_j \leq D_j^{class}$ where D_j^{class} denotes the maximum delay for a packet of class C_j for $1 \leq j \leq n$.

The expected delay of class j traffic is not more than the maximum delay tolerable by class j traffic requests.

A_2: $(\sum_{j=1}^{n} \lambda_j) + \lambda_i^p \leq F * \mu_{avg}$ where F is the system utilization factor and $0 < F \leq 1$.

The demanded service rate is not more than the service rate offered by the system.

However, these estimates are pessimistic because the peak rate of the service request is used to compute the delay and the load on the system. To make the call admission control algorithm optimistic, rather than dropping a call when it does not satisfy either of the conditions A_1 or A_2 drop it with a probability P_{drop}. The following section explains the performance of the call admission control algorithm for varying values of P_{drop} under different traffic mix.

5.4 Simulation Model

The call admission algorithm is studied $w.r.t$ the ratio of the number of calls dropped to the calls arrived and the ratio of the number of frames during which there are

delay violations to the total number of frames during the simulation. For the simulation setup assume four classses of service and the delay requirement of each class is tabulated below:

Class	Delay allowed in frames
C_1	2
C_2	10
C_3	20
C_4	50

In order to ensure that the sources confirm to their corresponding peak rate that is declared at call admission time we use the leaky bucket traffic shaper at each source. Parameters for the leaky bucket model:

- Bucket size is fixed to a value in the range 100 to 300 packets.
- The ratio of the leak rate and the maximum rate at which packets leave the source lies in the range 0.1 to 0.5.
- The observation period based upon which the peak rate is determined $\delta_t = 500$ frames. For simulation each frame is assumed to be of 100 msec duration.

Each simulation experiment was conducted for 10,000 frames and the value at each data point is obtained as the average of three simulation runs.

5.4.1 Call Requests Model

Call arrival and call holding time are governed by a poisson process. The call arrival rate is denoted as λ calls per frame. Arrival of calls of each class are equally likely. The call holding time denoted by $\mu_{hold} = 5$ minutes (3000 frames). Each source is

assumed to have an active time and idle time. The active time is determined by the time taken to send a file. To simulate the active time we use Pareto distribution which gives the size of a file to be transmitted. The average size of the file is fixed to 6 packets and $\alpha_{pareto} = 1.2$. The idle time is governed by Weibull distribution with the parameters $k_{weibull} = 0.5$ and $\theta_{weibull} = 3.5$. The service rate of the system is adjusted after $N = 100$ frames.

To study the performance of the call admission control algorithm we first execute the simulation experiments for the case in which the traffic has only two classes, P_{drop} is fixed. For each experiment we collect the following data:

(i) the ratio of the number of calls dropped to the number of arrived calls i.e., the ratio of call rejections,

(ii) and for each class the ratio of the number of frames where there was a delay violation to the total number of frames during the simulation period.

For the simulation experiments the utilization factor F was fixed at 0.8.

5.4.2 Results for two class traffic mix

Figures 5.2-5.6 are the results obtained with two classes of traffic in the system. Figure 5.2 shows the performance of the CAC algorithm *w.r.t* the change in the number of calls dropped for varying arrival rate (λ), different class mixes in the arrival traffic, varying P_{drop} and varying airlink frame loss probability. As per the simulation setup the call holding time is 3000 frames i.e., the service rate is $\frac{10}{3} * 10^{-4}$ calls per frame. Since each call has an idle time and active time the system accomodates additional call requests during the idle period of the call. Hence, the system does not drop most of the calls even when the call arrival rate (λ) is more than $\frac{10}{3} * 10^{-4}$. With

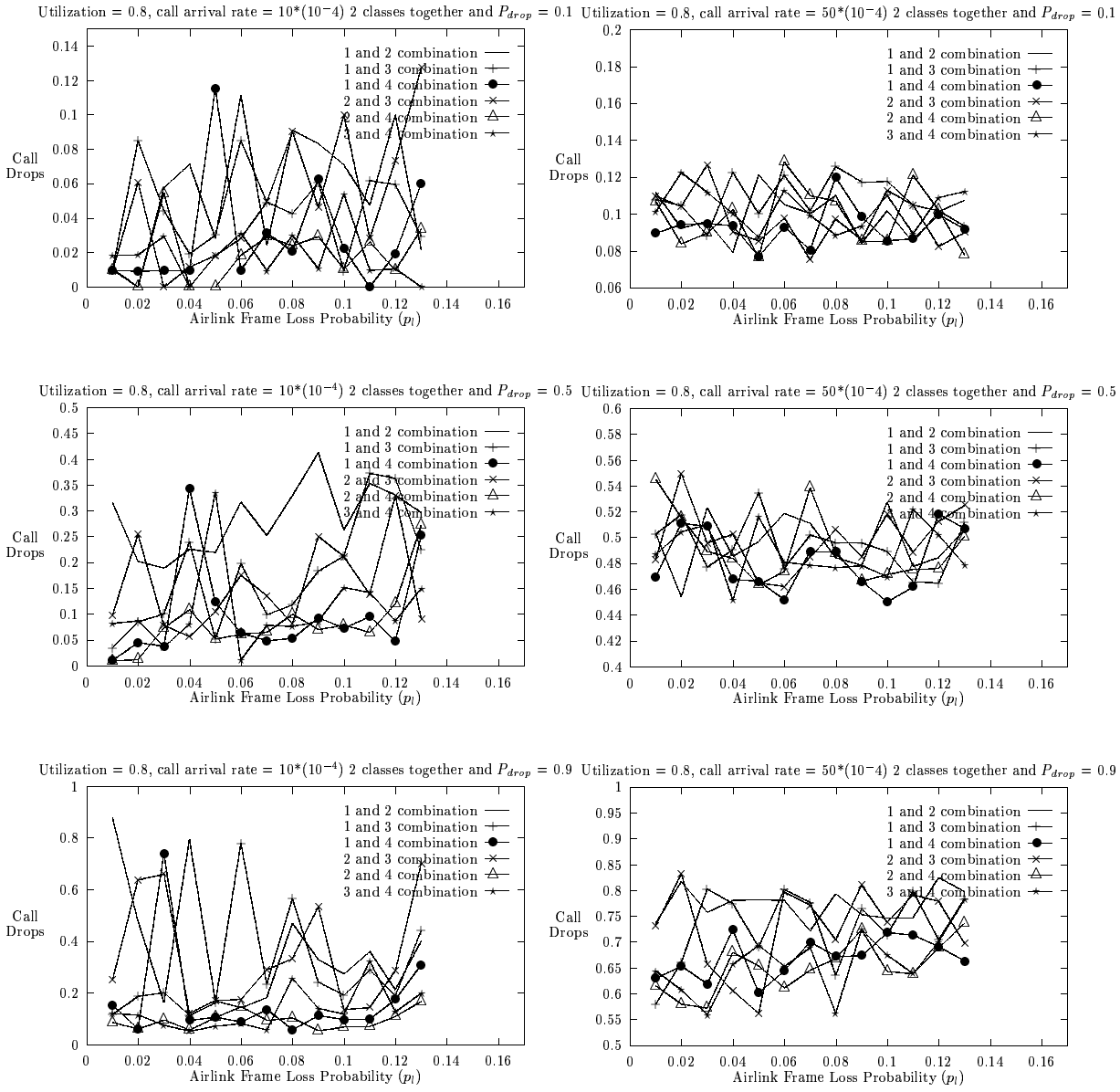

Figure 5.2: Calls dropped for a traffic mix of two classes

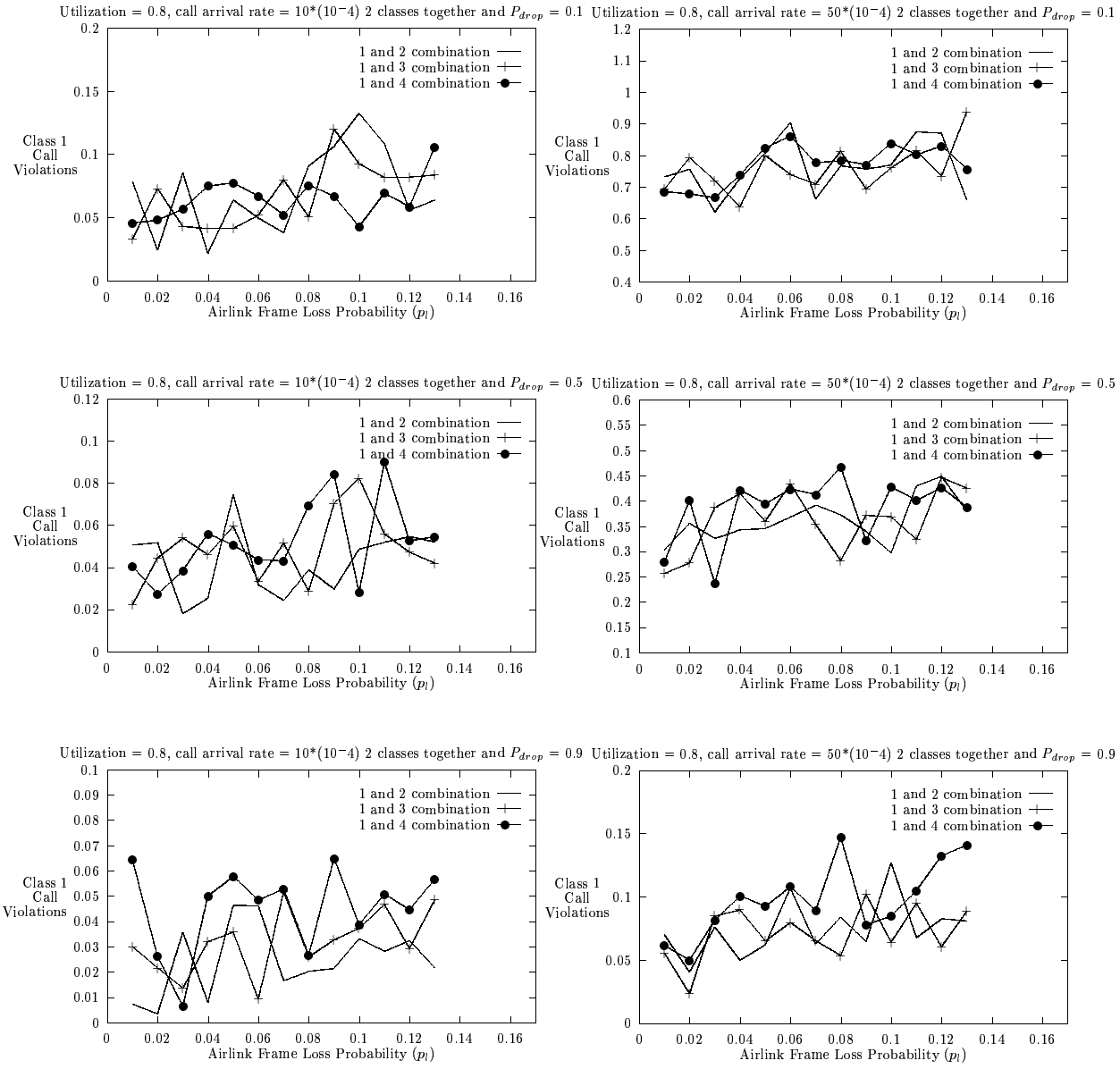

Figure 5.3: Ratio of Call violations for Class 1 traffic in a traffic mix of two classes

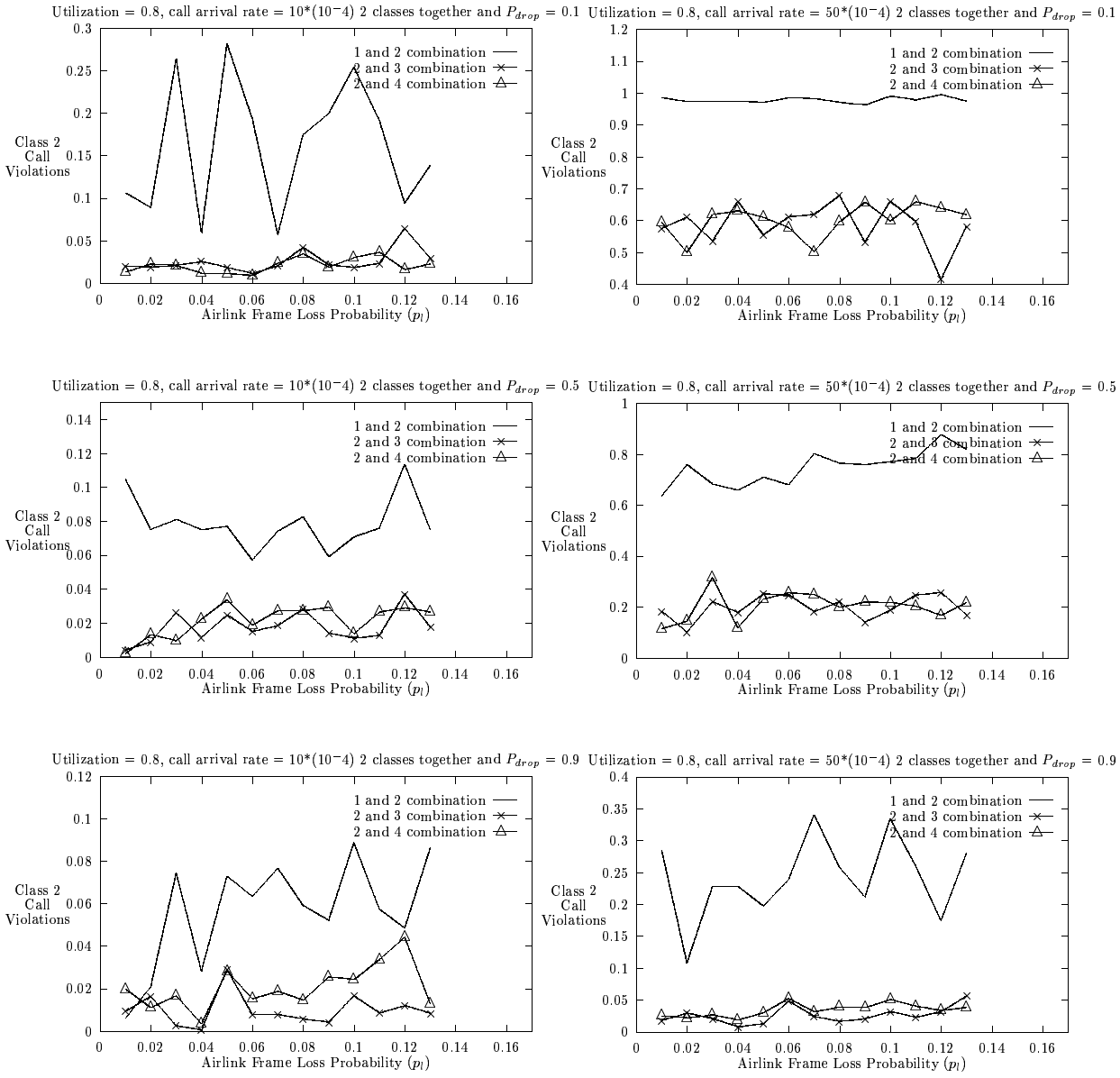

Figure 5.4: Ratio of Call violations for Class 2 traffic in a traffic mix of two classes

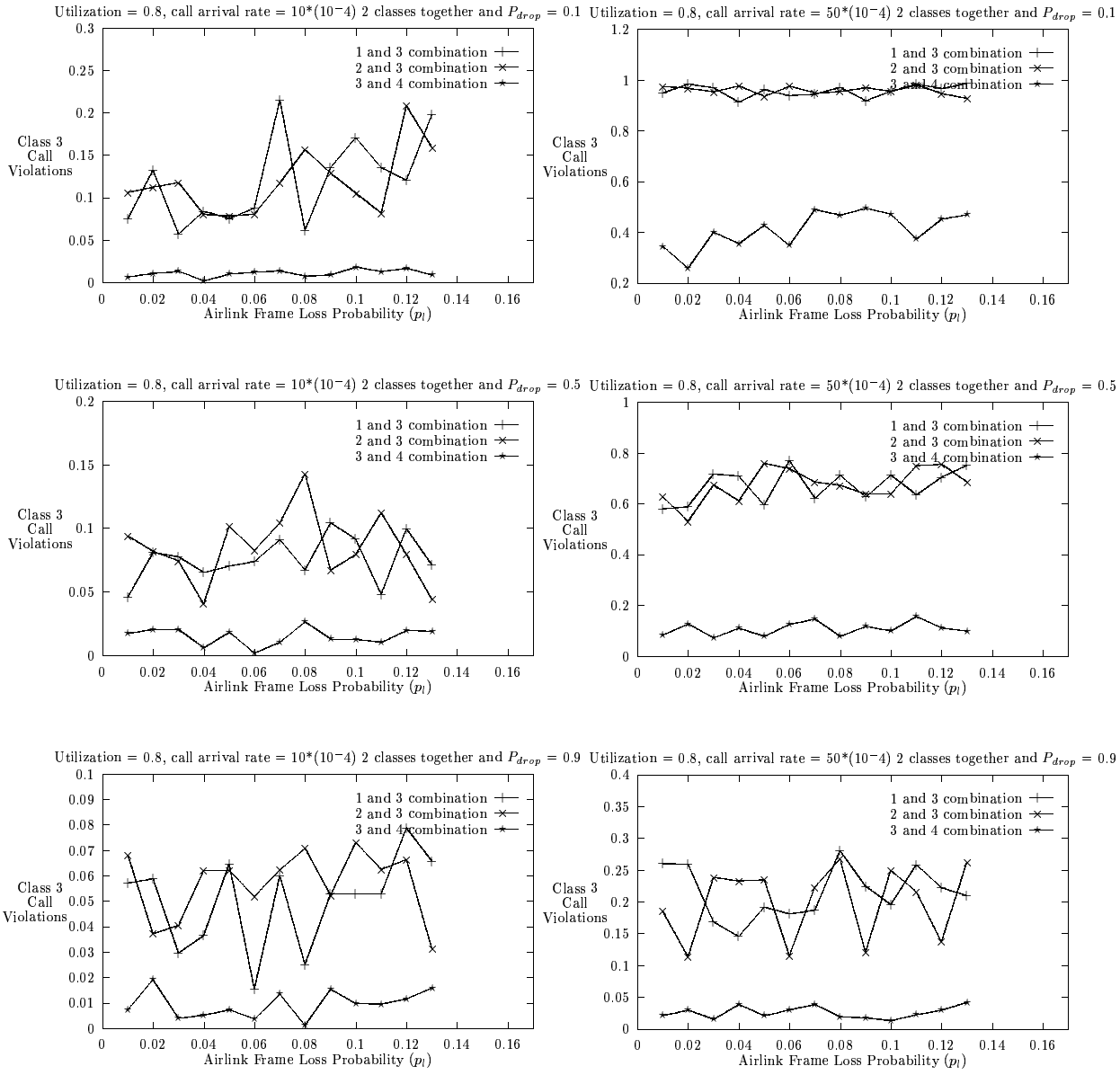

Figure 5.5: Ratio of Call violations for Class 3 traffic in a traffic mix of two classes

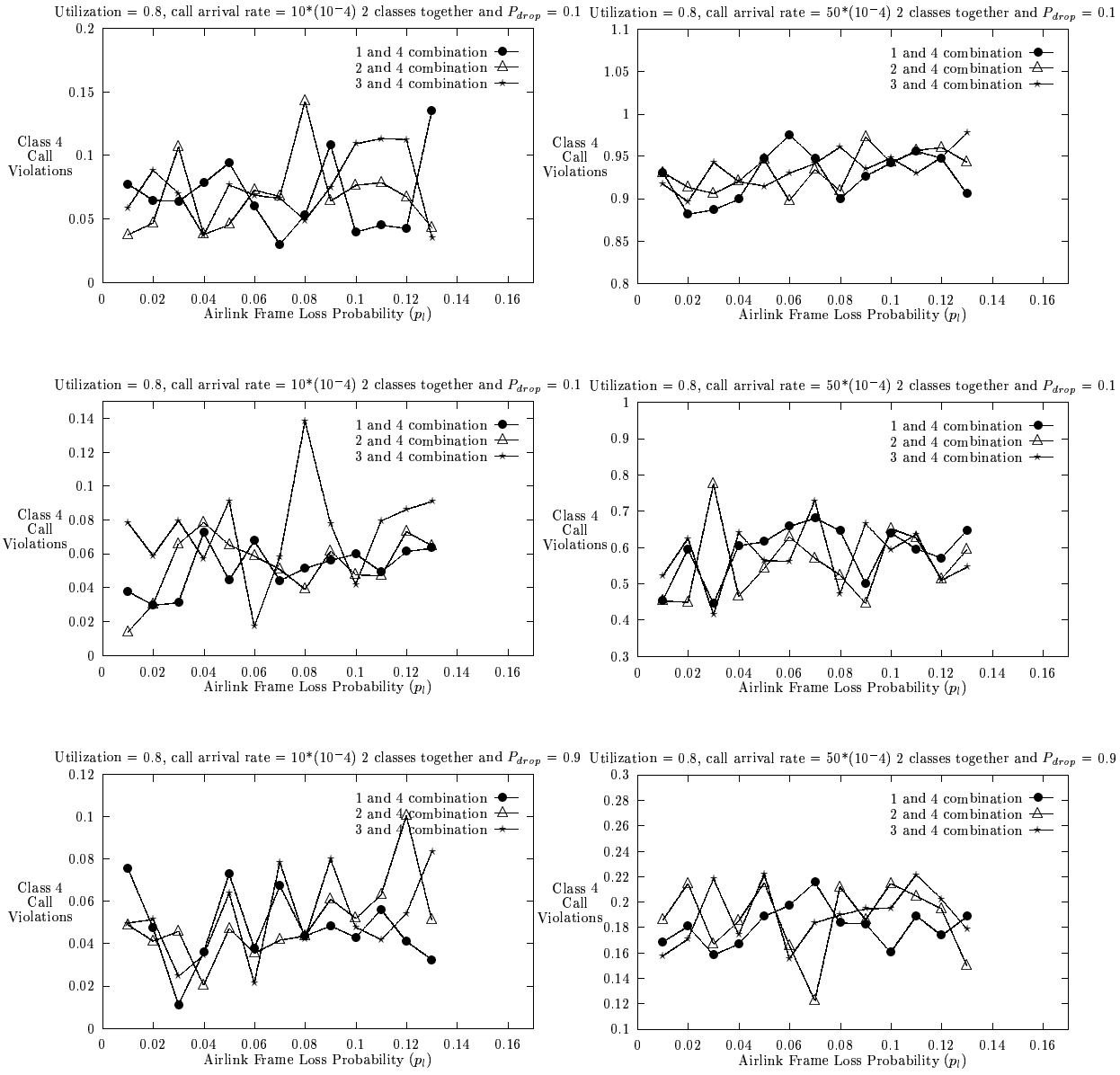

Figure 5.6: Ratio of Call violations for Class 4 traffic in a traffic mix of two classes

an increase in the arrival rate from $10*10^{-4}$ to $50*10^{-4}$ the ratio of calls dropped is bounded by P_{drop}. This is because with an increase in the arrival rate most of the calls if admitted will lead to a violation of one or both of the call admission requirements A_1, A_2, and when calls violate either of the call admission conditions they are dropped with probability P_{drop} so the ratio of calls dropped is bounded by P_{drop}. Even with an increase in P_{drop} the ratio of calls dropped depends upon the class mix. Consider a traffic mix of class 1 and class 2. Class 1 and class 2 are the first and the second classes which come up when the classes are ordered in increasing order of their delay requirement. For this traffic mix even at an arrival rate of $10*10^{-4}$ the ratio of calls dropped is bounded by P_{drop}. On contrary consider a class mix of class 3 and class 4, at an arrival rate of $50*10^{-4}$ the ratio of calls dropped is much lower than P_{drop}. Hence, more calls can be admitted into the system. This clearly shows that for different traffic mix there has to be a different P_{drop} if the system is to be used to its full capacity. With an increase in the airlink frame loss probability the number of calls dropped is bounded by P_{drop}. This is because with an increase in the airlink frame loss probability the service rate of the system reduces so the new calls violate the condition A_2 and their admission into the system is controlled by P_{drop}. There are oscillations in the number of calls dropped, this is because of the differences in the QoS requirements of traffic requests in the traffic mix. If the traffic requests in the traffic mix have a large difference in their QoS requirements, then for the same call arrival rate if there were more calls of lower priority traffic request during the simulation then they would be fewer number of calls dropped as compared to the situation where there are more call requests of higher priority class traffic requests.

Figure 5.3 shows the variation of the ratio of the number of simulation cycles during which a class 1 call had a delay violation to the total number of simulation

cycles for the experiment. Since packets are scheduled as per the priority of the classes, and class 1 calls are given the highest priority, so the ratio of call violations for a class 1 traffic request is not controlled by the traffic mix. With an increase in P_{drop} more calls are dropped, hence there is a decrease in the ratio of call violations (fewer call requests competing for resources). However, an increase in P_{drop} results in fewer calls in the system and this translates to lower revenue. Hence, there is a tradeoff with the number of calls dropped (revenue) and the call violations (QoS guarantee) which is controlled by the parameter P_{drop}.

Figure 5.4 shows the behaviour of the ratio of call violations for class 2 traffic requests for different traffic mix, change in P_{drop}, change in call arrival rate and for different airlink frameloss probability. Since class 2 has lower priority than class 1, for a traffic mix with class 1, class 2 calls have more number of violations as compared to class 2 call violations. For a traffic mix with either class 3 or class 4. As in the case of class 1 with an increase in P_{drop} since more number of calls are not admitted into the system even when the admission criterion are not met, there are fewer class 2 call violations. For a traffic mix with class 3 or class 4 since they have lower priority than class 2 call requests class 2 calls are not effected and the call violations are because of additional calls in the system (due to P_{drop}). Note that P_{drop} is the probability of dropping a call even when either of the conditions A_1 or A_2 are not satisfied, i.e., $1 - P_{drop}$ is the probability of admitting the call request. Figure 5.5 shows the variation of call violations of class 3 calls. Like class 2 calls, class 3 calls are not effected by the lower class calls. With an increase in P_{drop} which results in lower number of calls being admitted into the system even when they violate conditions A_1 and A_2, the number of call violations for class 3 traffic goes down. However, for a given value of P_{drop} the ratio of call violations of class 3 traffic is lower than the

ratio of call violations for either class 2 or class 1 traffic when combined with lower class traffic. This is because the delay constraints for class 3 traffic is not as severe as the delay constraint of class 1 or class 2 call requests. The results in Figure 5.6 shows the variation of call violations of class 4 traffic requests with varying P_{drop}, call arrival rate (λ), airlink frameloss probability and for different traffic mix. For all traffic mixes with class 4 calls, class 4 calls are effected the worst as this traffic is given the least priority.

5.4.3 Results for three class traffic mix

Figures 5.7-5.11 shows the results for experiments with three classes of traffic in the system. Figure 5.7 shows the variation of the ratio of calls dropped for different traffic mixes, varying airlink frameloss probablity and varying P_{drop}. Since the traffic has three classes of traffic and a call can be of any of the three classes with equal probability, for a given P_{drop}, the call drops for a traffic mix of three classes is lower than the call drops for a traffic mix of two classes. With an increase in the call arrival rate, for a given P_{drop} the call drops are equal for a traffic mix of two classes and for a traffic mix of three classes. This is because, the system is already overloaded so the traffic mix has no effect on the system performance. For a lower call arrival rate, with an increase in airlink frameloss probability (p_l), there is an increase in the number of calls dropped for a traffic mix of higher priority classes and for a traffic mix of lower priority classes there is no change in the number of calls dropped. For a traffic mix of higher priority classes each call has a low interpacket delay. It is most likely that the system will reach an overloaded state faster than in the situation where users can tolerate more delay hence they can be interleaved with more users. From the results in Figure 5.2 and 5.7 it is evident that different traffic mixes perform differently for a

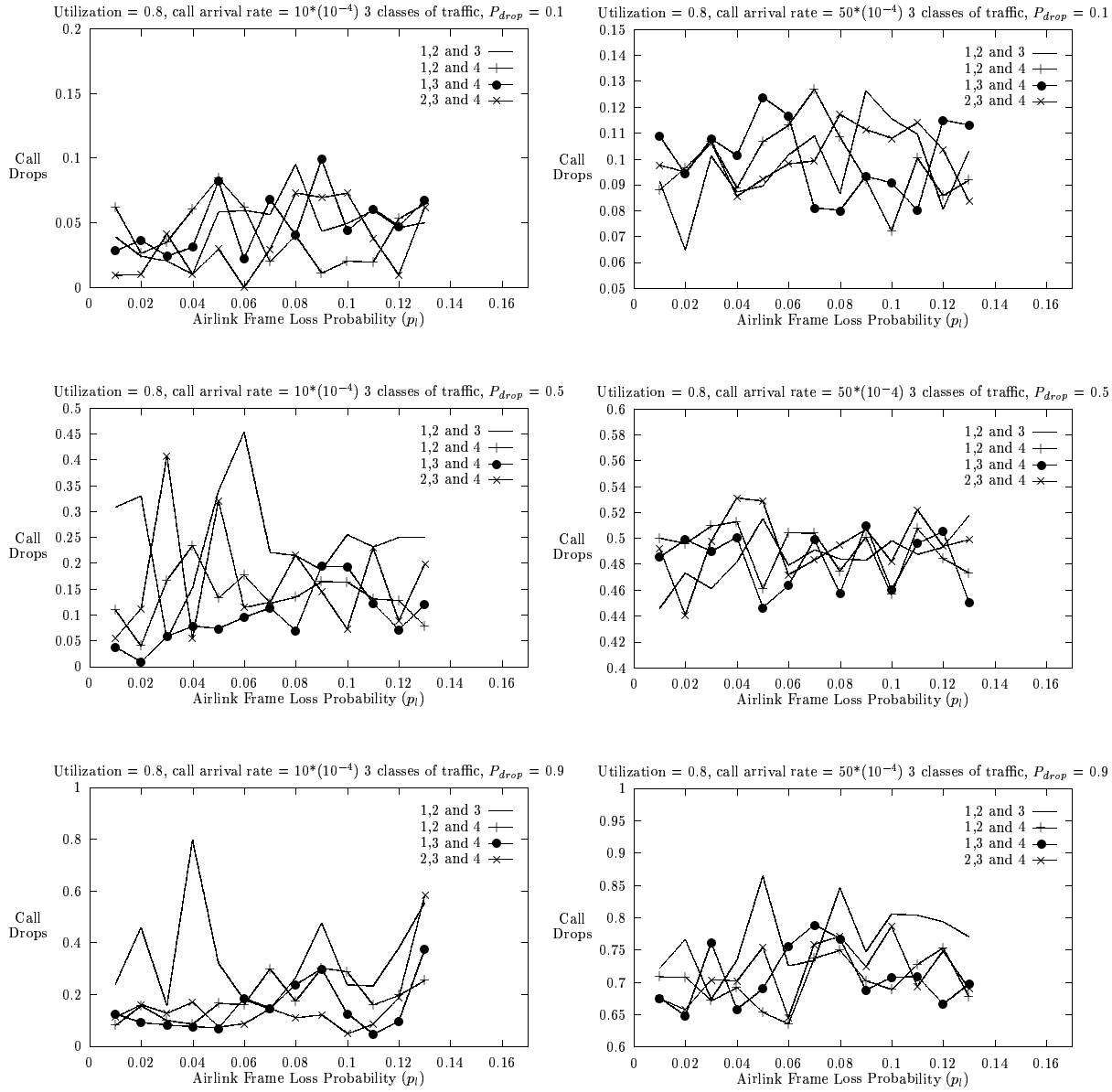

Figure 5.7: Calls dropped for a traffic mix of three classes

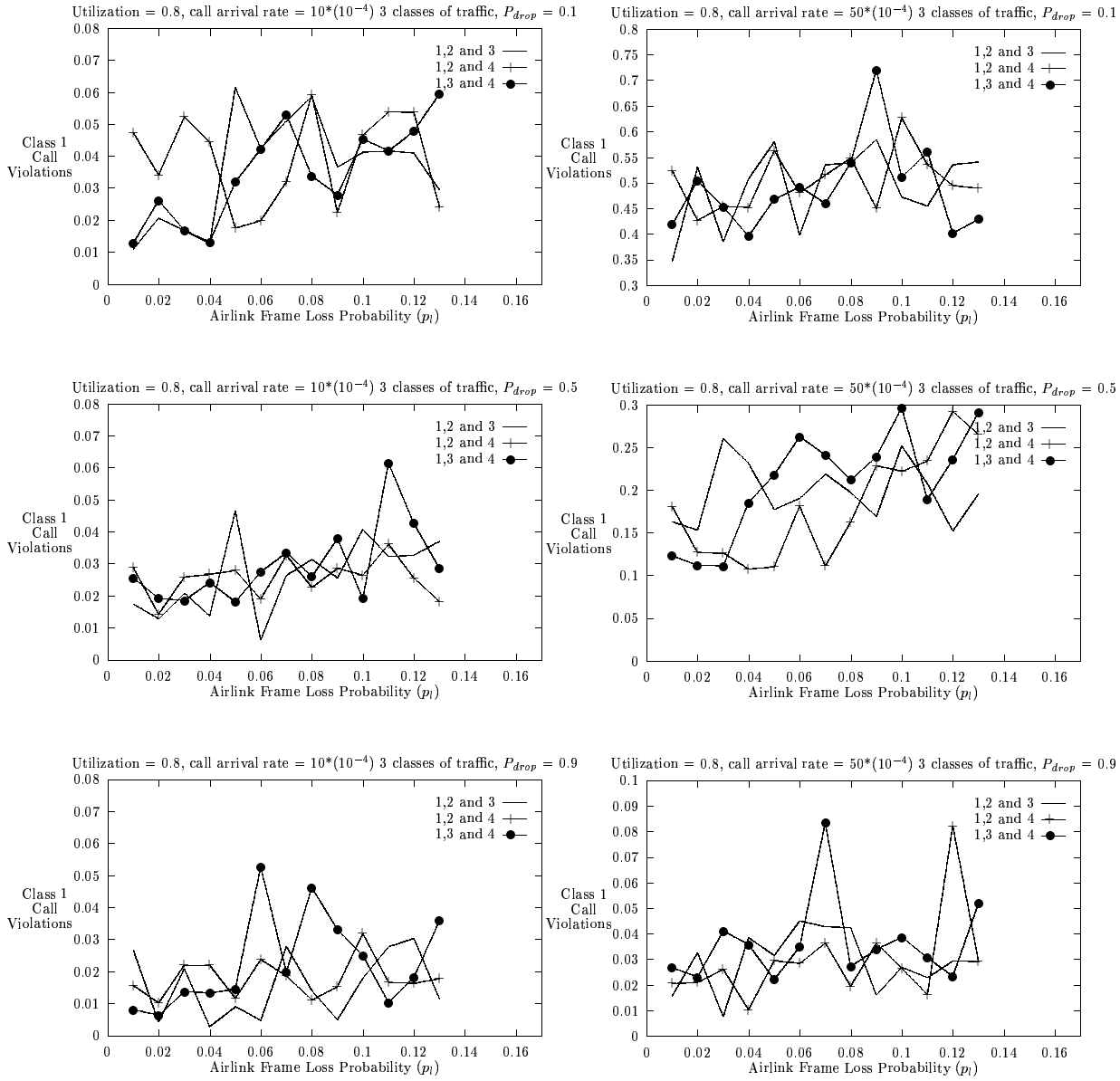

Figure 5.8: Ratio of Call violations for Class 1 traffic in a traffic mix of three classes

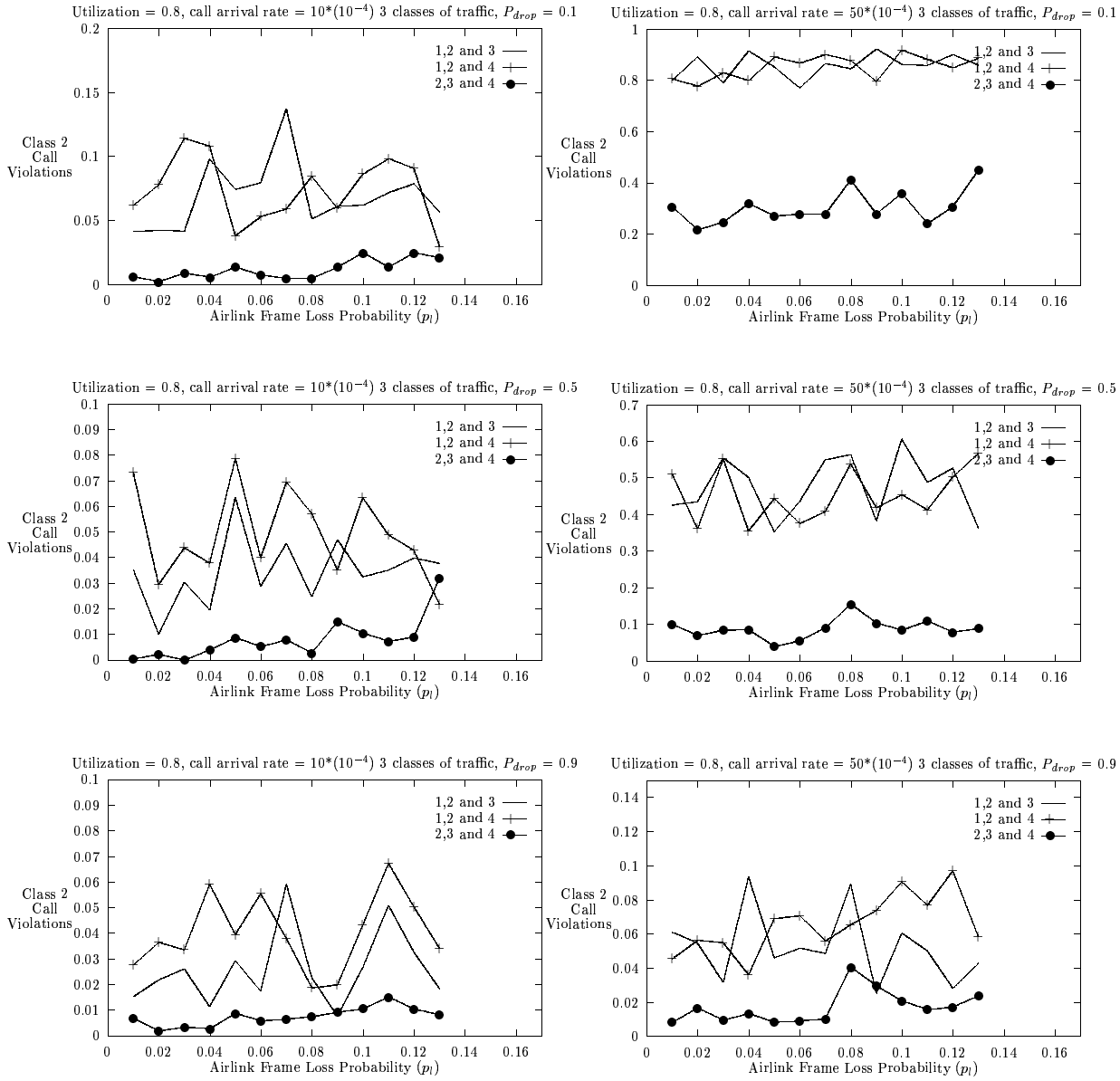

Figure 5.9: Ratio of Call violations for Class 2 traffic in a traffic mix of three classes

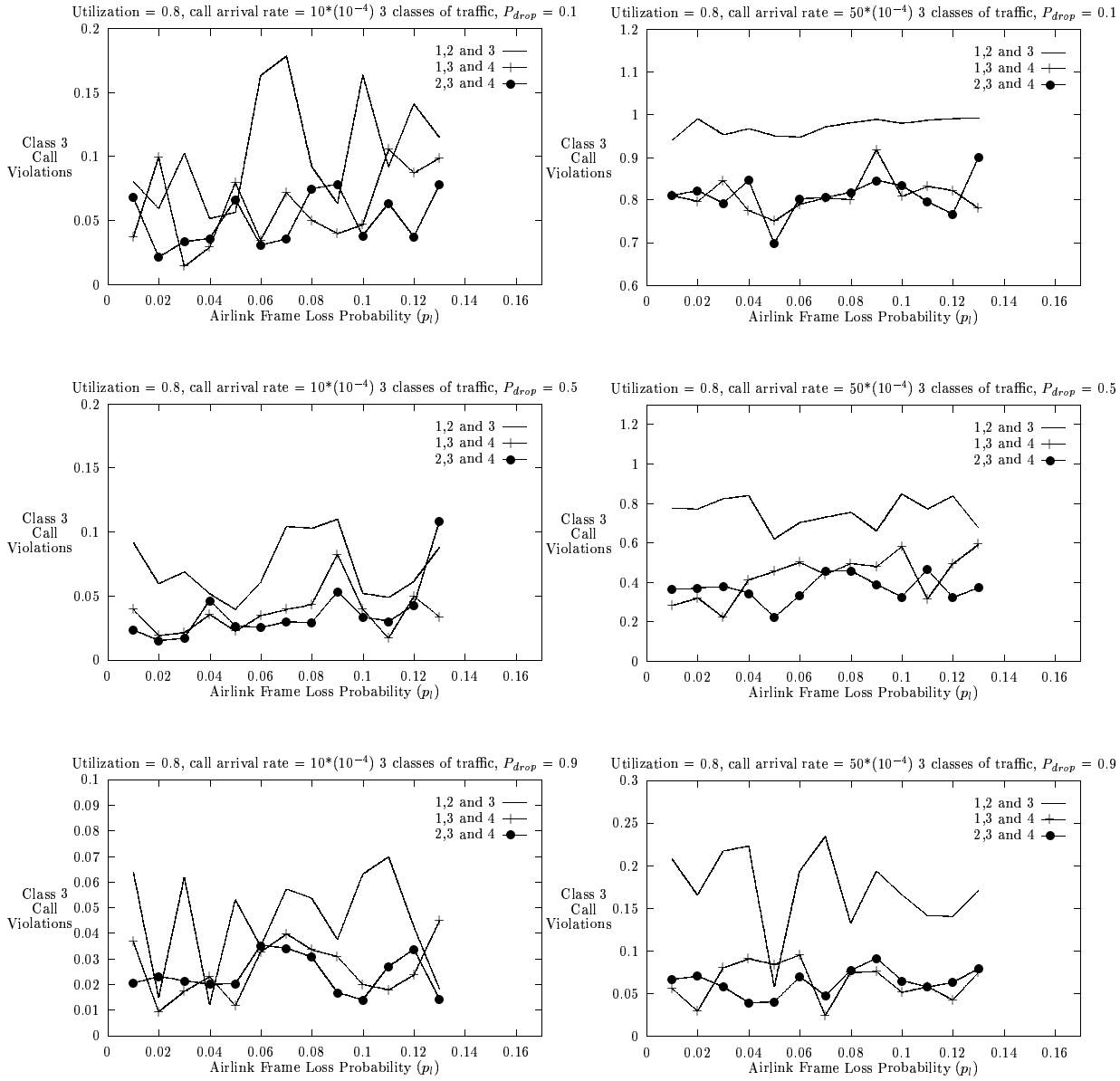

Figure 5.10: Ratio of Call violations for Class 3 traffic in a traffic mix of three classes

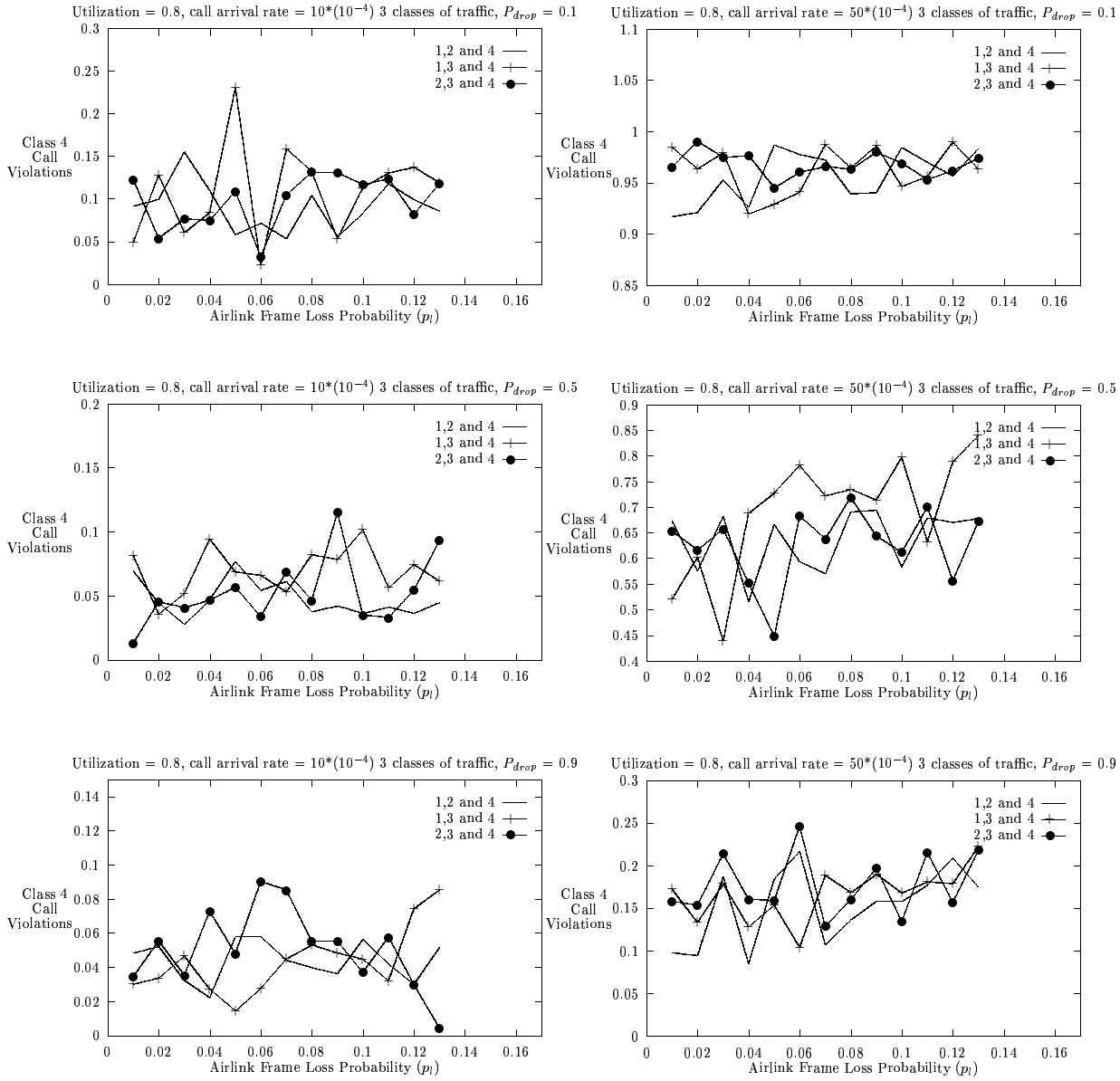

Figure 5.11: Ratio of Call violations for Class 4 traffic in a traffic mix of three classes

change in P_{drop}, and call arrival rate. Since the call arrival rate cannot be controlled by the system the system should adapt P_{drop} based upon the traffic mix and the current load on the system.

Figure 5.8 shows the variation of call violations of a class 1 call request. Since class 1 traffic is given higher priority the call violations of class 1 call requests is not effected by the traffic mix. However, when the results are compared to the results of call violations with two traffic classes in the system, the call violations for a traffic mix of three are fewer than the call violations for a traffic mix of two classes. This is because for a traffic mix of two classes a larger number of calls of the same traffic class compete for the resources as compared to the scenario for a traffic mix of three classes (a call can be of any of the classes with equal probability). Hence, the call violations for the higher priority class call requests decreases. When the system is overloaded the call violations increases with an increase in the airlink frameloss rate. When P_{drop} is higher an increase in airlink frameloss probablity (p_l) does not have much effect on the call violations because the call admission controls the load on the system. Again this is an optimization problem $w.r.t$ the number of calls dropped and the number of call violations.

Figure 5.9 shows the results of call violations for class 2 traffic requests. Since class 2 call requests have lower priority that class 1 call requests, for traffic mixes with class 1 call requests class 2 call requests have higher call violations as compared to the scenario where the traffic mix does not include class 1 traffic. For a traffic mix of three classes with class 1, class 2 requests the call violations for class 2 call requests are lower for the case when the third traffic class is class 3 as compared to the case when the third traffic class is class 4 requests. This is because for a traffic mix of higher class traffic more calls are dropped hence the likelihood of larger number of

class 1 calls getting admitted is lower in a traffic mix of class 1, class 2 and class 3 than in a traffic mix of class 1, class 2 and class 4. This means class 2 call requests have to compete with fewer higher prioity class requests hence the call violations are lower. However, the traffic mix has no effect for small values of P_{drop} and also with an increase in the call arrival rate. This is because in both cases the system is tuned to admit higher number of call requests so there will be larger number of call violations. Figure 5.10 shows the variation of call violations of class 3 call requests. Class 3 call request suffer the most in for a traffic mix with class 1 and class 2. Since class 1 and class 2 calls are higher priority call requests than class 3 requests, class 3 call request have to comepete with two higher priority calls hence there are more number of call violations for this traffic mix. For other traffic mix where class 3 calls have to compete with only one higher priority call request the call violations are identical. For lower call arrival rate i.e., for fewer number of incoming calls to the system, with an increase in P_{drop} there is not much difference in the call violations. This is because most of the incoming calls satisfy the call admission conditions A_1 and A_2. For class 4 call requests have the lowest priority, in every traffic combination they need to compete with two higher priority class call requests. Hence, the traffic combination has no effect on the call violations of class 4 call requests. The results for class 4 call violations are shown in Figure 5.11.

In the results presented the system uses the adaptive service rate algorithm to tailor the CAC algorithm for wireless links, the delay computation algorithm was used to overcome the unreliability in the source characteristics at the time of call admission into the system. To overcome the unreliabiity in the decision made by the system whether to admit a call or not, based upon the call admission conditions A_1 and A_2 the system uses P_{drop}.

5.4.4 Adaptive P_{drop}

The results presented in subsections 5.4.2 and 5.4.3 show that the quadratic delay estimate, the service rate adaptation alogorithm and P_{drop} control the number of calls in the system to meet the QoS promised to the users. The results presented were shown for different values of P_{drop} and for different traffic mix. From the results we infer that the value of P_{drop} should be varied for different traffic mix, hence, it is not possible to fix a value for P_{drop} as the traffic in the system changes dynamically. However, instead of having different value of P_{drop} for different traffic mix we can have different P_{drop} for different classes of traffic. Say for example if there are n number of classes of traffic in the system (in the simulation model $n = 4$), each class of traffic has a P_{drop} and for class i, let P_{drop} be denoted by P^i_{drop}. If P_{drop} is small then probabilistically more calls are admitted into the system even if they violate the call admission conditions. On the contrary if P_{drop} is high for lower traffic loads The system may drop more number of calls, this results in loss of revenue. The number of calls dropped and the ratio of call violations are the factors by which P_{drop} is adjusted to make the system work at an optimal point. The adaptation algorithm is executed after an observation period of N number (similar to service rate adaptation algorithm) of frames. During each observation period, number of calls arrived, calls dropped and the number of delay violations for each class of service are maintained. They are denoted as $Arrv_i$, $Drop_i$ and $Dviol_i$ respectively.

Algorithm for adapting P_{drop}

for $j = 1$ to n **do**

if $(\frac{Dviol_j}{N} > P^j_{drop})$ (Call violations are more than what P_{drop} is set to)

$P^j_{drop} = P^j_{drop} + \frac{\frac{Dviol_j}{N} - P^j_{drop}}{N}$

(increment P_{drop} to reduce the load on the system)

else

if ($Arrv_i > 0$) (there are call arrivals of class i

during the observation period)

if($\frac{Drop_i}{Arrv_i} < P^i_{drop}$) (ratio of calls dropped are fewer

than P_{drop})

$P^i_{drop} = P^i_{drop} + \frac{\frac{Drop_i}{Arrv_i} - P^i_{drop}}{N}$

reduce P_{drop} to admit more calls)

else (if there were no call arrivals)

$P^i_{drop} = P^i_{drop} - \frac{P^i_{drop}}{N}$

(reduce P_{drop} by a fraction)

$Arrv_i = 0$

$Drop_i = 0$

$Dviol_i = 0$

For a call to be admitted, the call admission conditions A_1 and A_2, have to be met. These conditions ensure that upon admitting the new call request there are no delay violations and the service rate demanded is not more than what the system canhandle. If either of the condition fails then a call of class i is dropped with probability P^i_{drop}. However, upon arrival of a new call request say of class i it might be possible that the delay requirements of class i are met but there is a violation of delay requirements of classes having lower priority than class i. Let there be a call request of class i, the excess delay for the calls of class j due to the arrival of the new call be denoted as $Excess_j$. For the new call to be admitted not only the excess delay of class i but the

excess delay of all the classes with lower priority have to be within a bound for the call to be admitted. For the call to be admitted the ratio $\frac{Excess_j}{D_j^{class}}$ has to be less than P_{drop}^j for all $i \leq j \leq n$. Moreover, to compute the load on the system (condition A_2) the peak rate of the new call request is considered, this leads to an over estimate on the system load. To compeansate for the over estimate on the system load compute the excess load on the system, denote it as $Excess_{sys}$ and for the call to be admitted the fraction $\frac{Excess_{sys}}{F*\mu_{avg}}$ is $< P_{sys}$ where P_{sys} is the probability of dropping a call due to violation on the load on the system.

The call admission conditions for a call arrival of class i are:

NA_1: $for\ 1 \leq j \leq n$

$Excess_j = D_j^{class} - \hat{D}_j$

$\frac{Excess_j}{D_j^{class}} < P_{drop}^j$

The ratio of the excess delay to the delay tolerable for that class Of traffic request is lower than P_{drop}^j

NA_2: $Excess_{sys} = F * \mu_{avg} - \left[\left(\sum_{j=1}^{n} \lambda_j\right) + \lambda_i^p\right]$

$\frac{Excess_{sys}}{F*\mu_{avg}} < P_{sys}$

The excess service rate demanded by the system to the service rate of the system Is not more than P_{sys}

With the two conditions NA_1 and NA_2 we admit to admit more calls into the system even when the system is overloaded. This is done to the extent to, optimize the number of users in the system and the reliability in meeting the QoS requirements of the users.

Algorithm to adapt P_{sys}

for $j = 1$ to n **do**

$S_{viol} = S_{viol} + Dviol_i$

$S_{drop} = S_{drop} + Drop_i$

$S_{arrv} = S_{arrv} + Arrv_i$

if $(\frac{S_{viol}}{n*N} > P_{sys})$ (n is the number of classes of traffic)

$P_{sys} = P_{sys} + \frac{\frac{S_{viol}}{n*N} - P_{sys}}{N}$

else

if $(S_{arrv} > 0)$

if$(\frac{S_{drop}}{S_{arrv}} < P_{sys})$

$P_{sys} = P_{sys} + \frac{\frac{S_{drop}}{S_{arrv}} - P_{sys}}{N}$

else

$P_{sys} = P_{sys} - \frac{P_{sys}}{N}$

The parameter P_{sys} has to be adjusted based upon the load on the system. When the ratio of the number of simulation cycles where there were QoS violations to the number of simulation cycles is larger than P_{sys} then the system increases the value of P_{sys}. This will reduce the number of calls admitted due to violation of condition NA_2. However, if the number of calls dropped are lower than P_{sys} then P_{sys} has to be reduced because the system is not overloaded.

5.4.5 Simulation Results

Figures 5.12-5.21 show the variation of the ratio of calls dropped for a traffic mix of four classes. These results compare the performance of the call admission algorithm with adaptive P_{drop} and call admission control algorithm with fixed P_{drop}. The results are presented for the cases when the system is not overloaded and when the system is

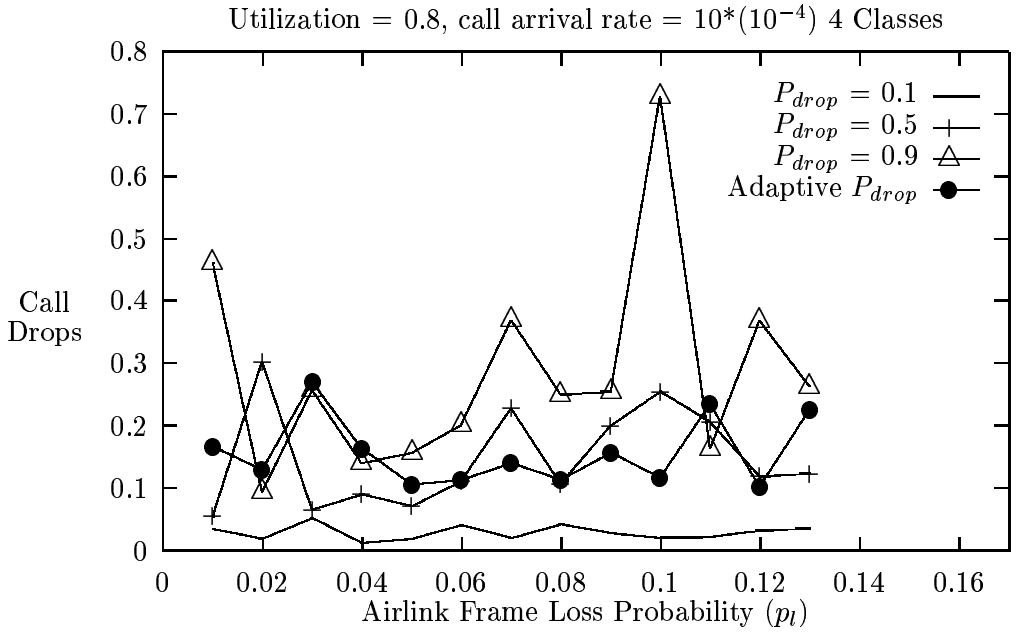

Figure 5.12: Calls Dropped when the system is not overloaded

overloaded. When the system is not overloaded the adaptive P_{drop} algorithm operates at the point where it admits most of the calls. It drops fewer number of calls as compared to the fixed P_{drop} algorithm where P_{drop} is set to 0.9. For the case when the system is overloaded (arrival rate $50*10^{-4}$) adpative P_{drop} algorithm drops more calls than the fixed P_{drop} algorithms. This is to meet the QoS demands of the call requests in the system.

Figures 5.14 and 5.15 plot the call violations for class 1 call requests. From Figure 5.14 it is observed that the adaptive P_{drop} algorithm does not have more call violations than for the fixed P_{drop} algorithm where P_{drop} is fixed to 0.9. Note that the call admission control algorithm using the adaptive P_{drop} algorithm drops less number of calls as compared to the call admission control algorithm with fixed P_{drop}. This verifies that the adaptive P_{drop} adapts to the system load and adjusts itself to meet

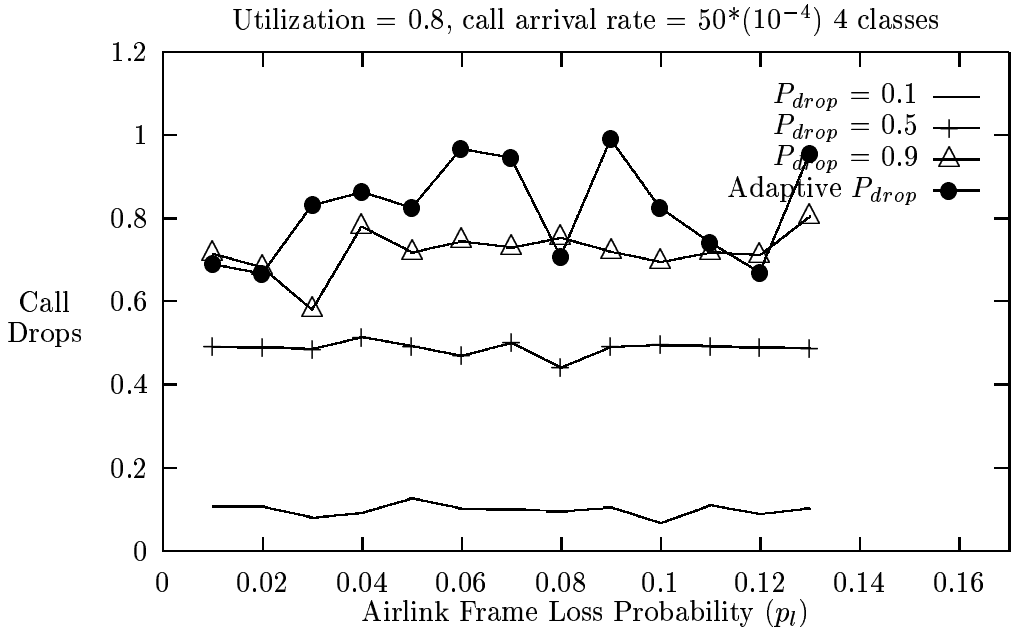

Figure 5.13: Calls Dropped when the system is overloaded

the QoS requirements and minimizes the number of calls dropped. For the overloaded case the adaptive P_{drop} drops more number of calls to ensure the QoS requirements of the existing calls in the system. Figures 5.16 and 5.17 plot the results for call violations for class 2 calls and Figures 5.18 and 5.19 plot the results for call violations for class 3 calls in the system. The results from these plots follow the same trend as the results in Figures 5.14 and 5.15. Figures 5.20 and 5.21 plot the call violations for class 4 call requests. The observations made in these results are similar to the observations made for class 1, class 2, and class 3 call violations. In addition to that, as compared to the performance of fixed P_{drop} algorithm traffic classes of lower priority do not have larger QoS violations to ensure that the QoS requirements of higher priority class are met. This is acheived with an increase in the number of calls dropped but this ensures that users are ensured their QoS requirements promised to them at the time when the call was admitted to the system.

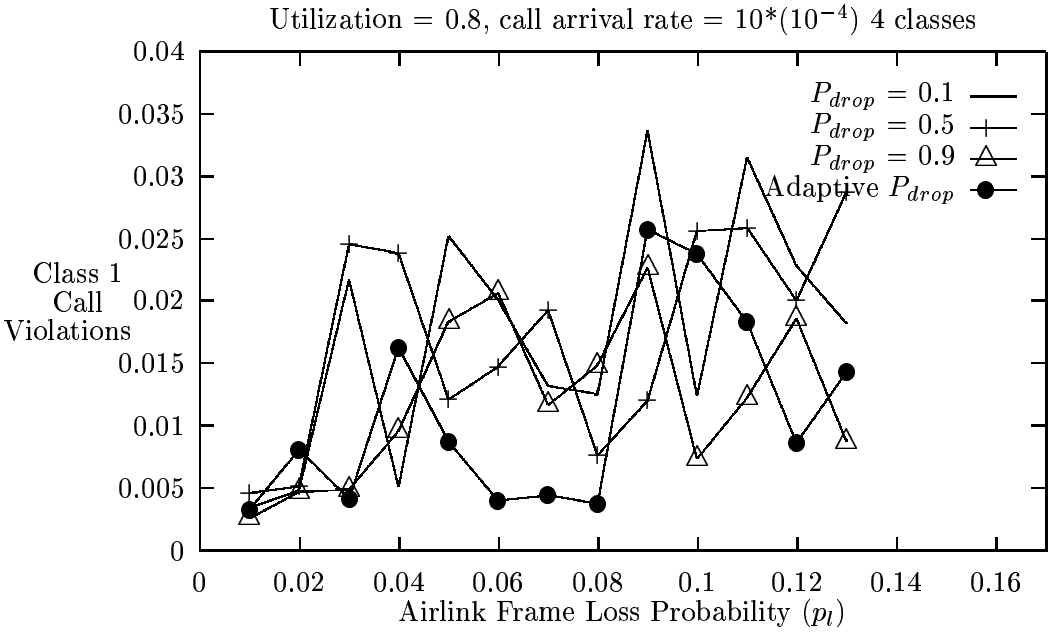

Figure 5.14: Class 1 call violations when the system is not overloaded

5.5 Summary

In this chapter we presented an adaptive Call Admission Control (CAC) algorithm to control the traffic in the system and to meet the QoS promised to call requests at call setup time. The proposed CAC algorithm estimates the statistical queuing delay if a call is admitted by using a quadratic function which is obtained by fitting the observed statistical delay and the delay computed using the M/M/1 model. To prove the flexibility of the algorithm experimental results were presented for traffic scenario which do not conform to M/M/1 model. The service rate of the system and the reliability in the system estimate which is controlled by P_{drop} are adapted based upon the system performance and the deviation from the estimate. From the experimental results it was observed that the relaibility in the delay estimated varies for different classes of traffic requests, call arrival rates, traffic mix and the airlink

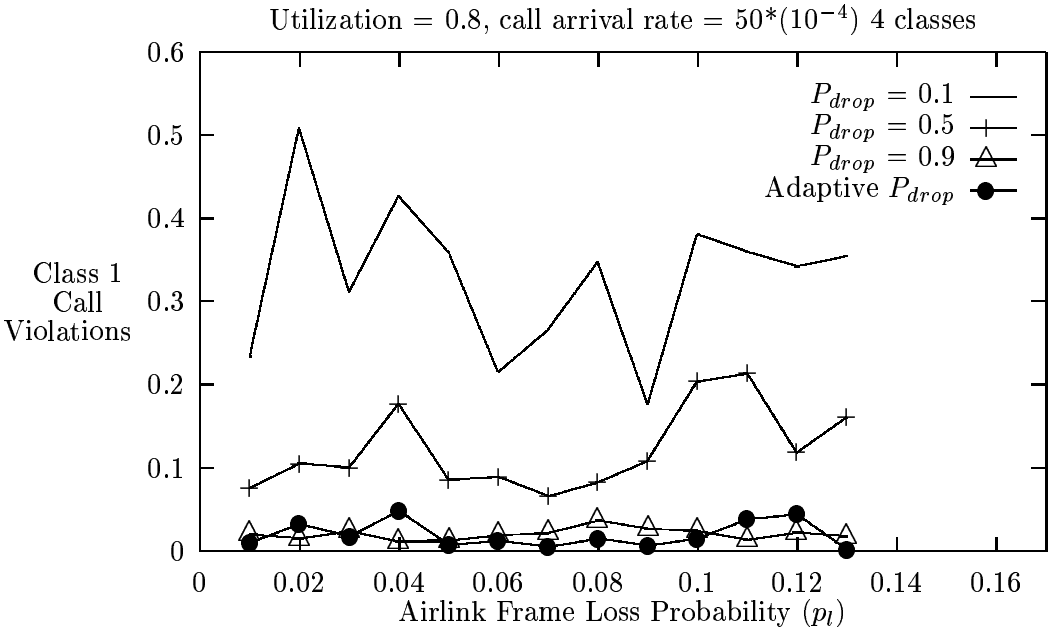

Figure 5.15: Class 1 call violations when the system is overloaded

frame loss rate. For this reason we associate different relaibility factor for different class of traffic request and we use this to ensure that the excess delay predicted is within the reliability limit.

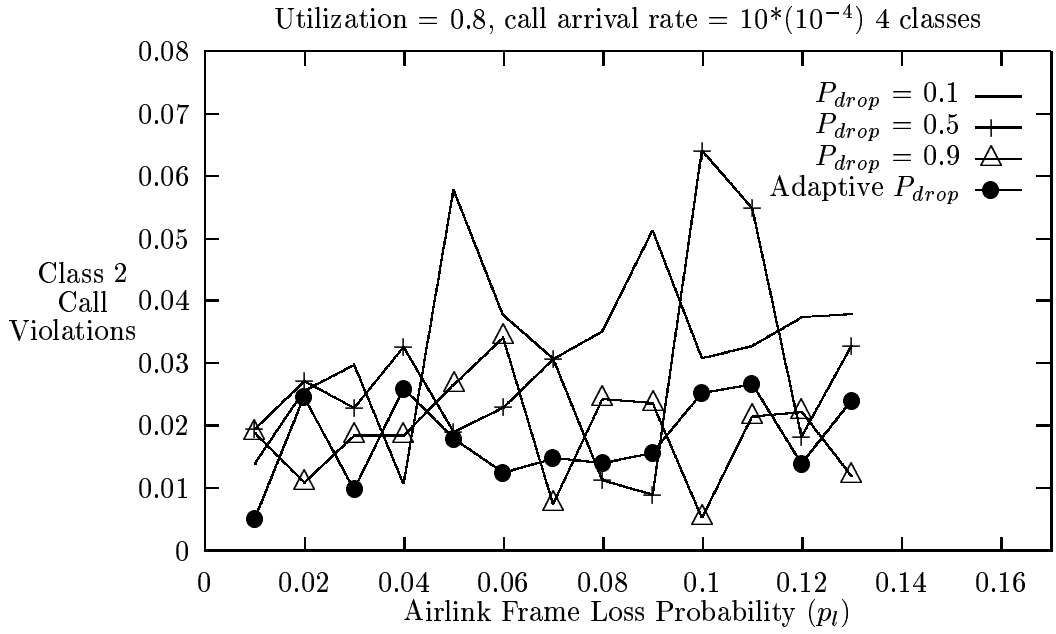

Figure 5.16: Class 2 call violations when the system is not overloaded

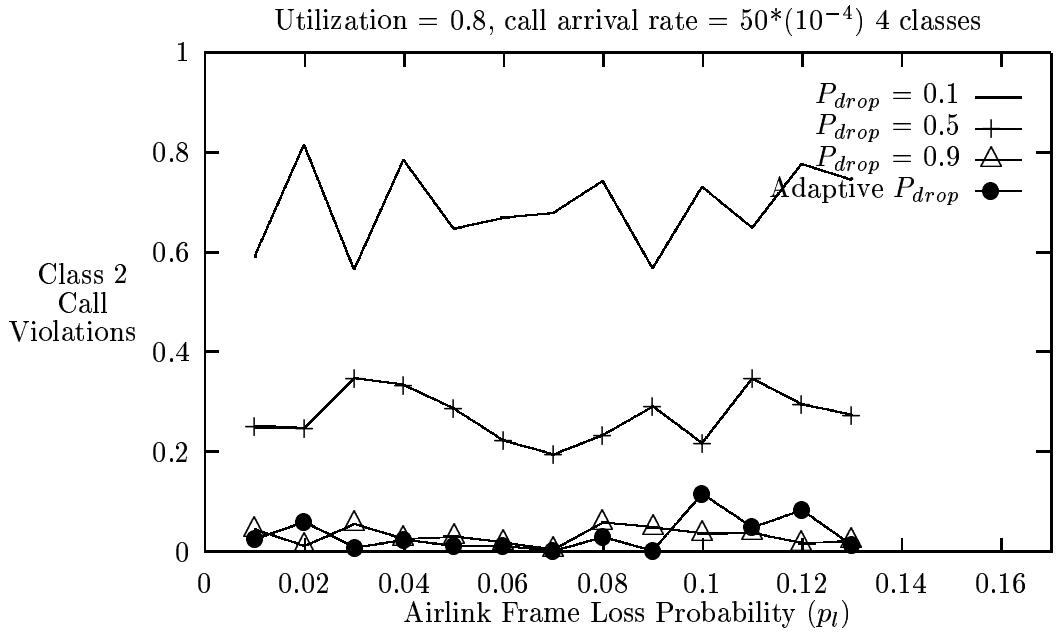

Figure 5.17: Class 2 call violations when the system is overloaded

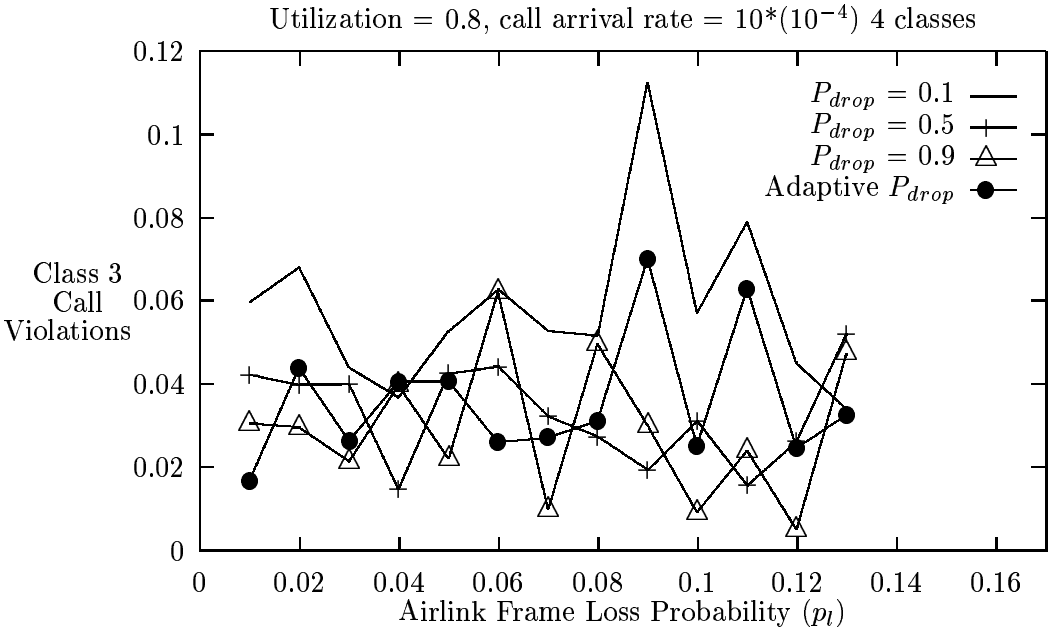

Figure 5.18: Class 3 call violations when the system is not overloaded

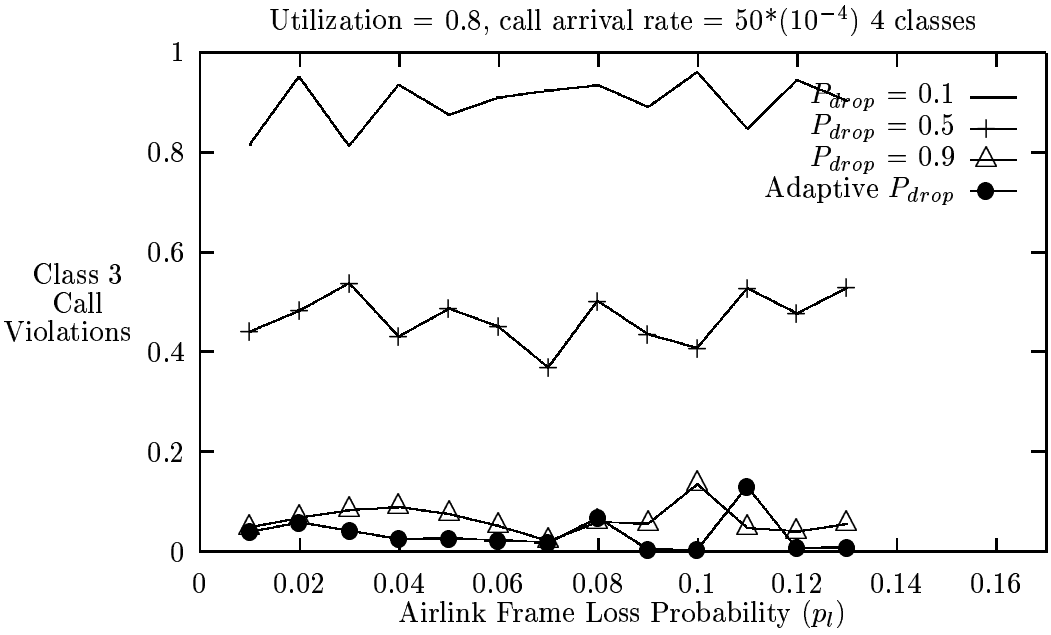

Figure 5.19: Class 3 call violations when the system is overloaded

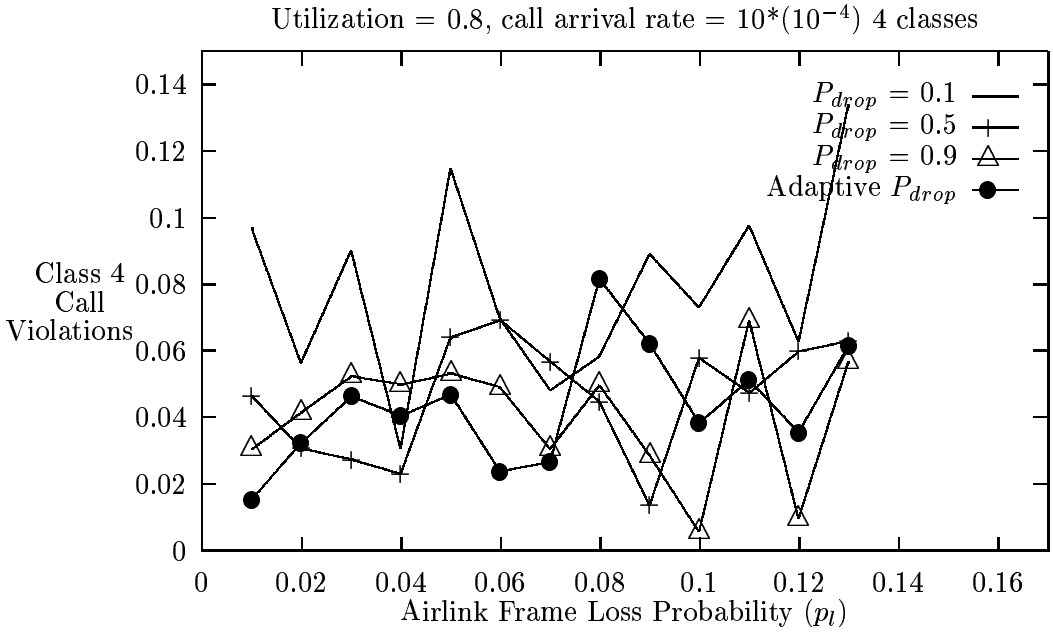

Figure 5.20: Class 4 call violations when the system is not overloaded

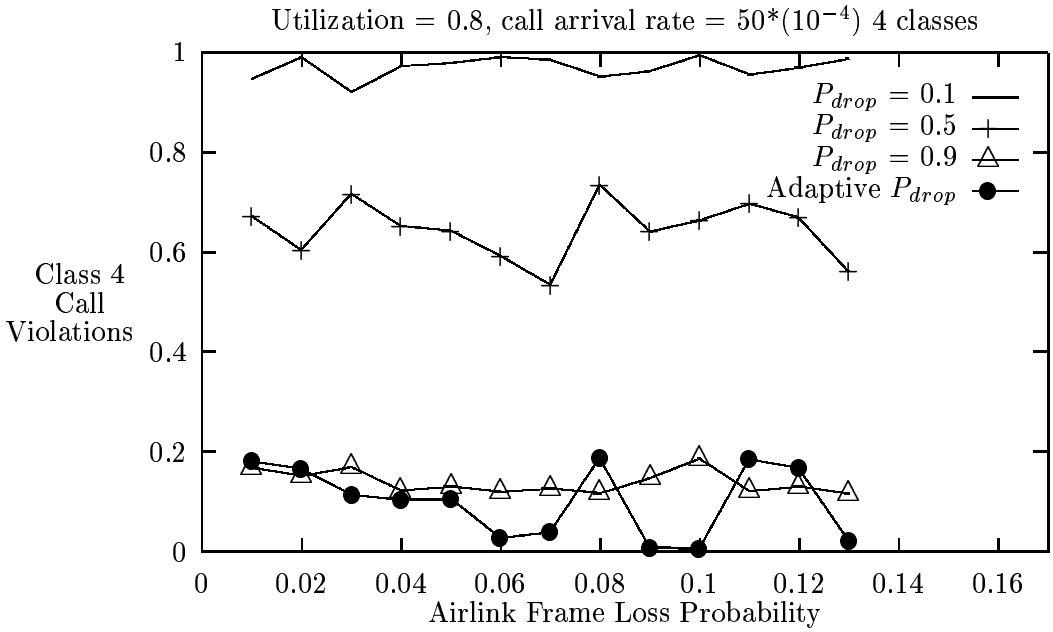

Figure 5.21: Class 4 call violations when the system is overloaded

CHAPTER 6

CONCLUSIONS

In this dissertation, we proposed algorithms for efficient bandwidth utilization and QoS provisioning in wireless networks. The algorithms presented propose a new frame structure and a call admission control mechanism which modify the frame structure and control the number of users in the system.

6.1 Adaptive Frame Structure

We presented a new frame structure which dynamically controls the number of users who are alloted to each timeslot, and the number of slots allocated to Mode-1 and Mode-2 traffic requests. User requests which are multiplexed at each timeslot are classified as Mode-1 traffic requests and Mode-2 user requests are multiplexed over the frame. The number of users supported by the system, the throughput of the system are the criteria which control the number of Mode-1 and Mode-2 requests in the system. Mode-1 traffic requests data is sent in each frame (requests are multiplexed over a timeslot in each frame) to have a bound on the inter-packet delay for a given frame loss probability.

A timeslot is divided into fractional timeslots called minislots, the bandwidth of each minislot is adjusted to carry the application with minimum bandwidth requirement. To meet the delay requirements for Mode-1 requests a few minislots in each slot are set aside to retransmit lost packets. These minislots are called retransmission minislots. In a timeslot the number of minislots to transmit user data and the number

of retransmission minislots are adjusted based upon the airlink frame loss probability. The retransmission list is modeled using a markovian model. The time taken to send a packet is computed as a function of the ratio of the number of retransmission minislots to the number of user data minislots and the airlink frame loss probability. The retransmission queue length (upper bound) is bounded to obtain a closed form solution for the expected number of retransmissions per user minislot.

Since all the Mode-1 requests can send packets in the same frame (in their minislots) to increase the number of users supported the number of slots allocated for Mode-1 transmission have to be increased. This is done at the cost of increased overheads (loss in throughout) as each user has to send his header information for data to be demultiplexed at the other end. Hence, the system has to adjust the number of users and the system throughput to operate at a point where the system performance is optimal. We define a QoS function (f_{QoS}) which is the product of the number of users supported and the system throughput. The operating point of the system is determined by optimizing the QoS function $w.r.t$ to the number of slots to be allocated to Mode-1 requests and number of slots to be allocated to Mode-2 requests. Note that this formulation is optimized for a given airlink frame loss probability and a fixed ratio of the header bits to the data rate of each slot. In real networks the number of header bits are constant but the airlink frame loss probability is not constant. For the system to operate at the optimal point with a change in airlink frame loss probability the system has to change the operating point. This is done by changing the frame structure, and the number of slots allocated to Mode-1 and Mode-2 requests. A change in the frame structure and a change in the number of timeslots allocated to Mode-1 and Mode-2 slots is achieved by changing the number of retransmission and user minislots in each slot and reallocating existing user requests of a particular

class of call requests to the slots of the same class to obtain empty slots. The empty slots obtained in the process are used to allocate new call requests of a class different from the class of call requests it was carrying earlier. The slots which have to be reallocated from Mode-1 to Mode-2 or vice versa are chosen based upon the current bandwidth of the slots being used. If R slots have to be reallocated from one class of call requests to the other class, the slots carrying the traffic of the class of call requests which has to be reallocated are sorted in increasing order of the available bandwidth and the last R slots of the sorted slots are chosen to be reallocated.

The performance of the new frame structure is compared with the performance of IS-136 system. The performance metrics being the throughput of the system, number of users carried by the system and the reliability in meeting the delay requirements. The new frame structure suffers $w.r.t$ the throughput of the system but it improves on the number of users carried by the system and the reliability in meeting the delay criteria for Mode-1 requests. The drop in the throughput of the system is because a timeslot is split into minislots and each minislot can carry one user request but he has the same amount if header information as he has when he is using the whole slot. There is an increase in the number of users supported by the system this is because the minimum slot duration is set based upon the minimum data rate application that the system can expect. This means more revenue for the system operator. The reliability in meeting the delay requirements promised at call setup time is because of the use of the retransmission minislots and the adaptive nature of the system to adjust the retransmission minislots based upon the airlink frame loss probability.

By setting the minimum slot duration to the minimum data rate application the system minimizes the amount of time for which the bandwidth allocated to a user is not used. Although this is done by increasing the number of overhead bits the system

is not idle for most of the time and the good-put (amount of data sent by the system) of the system increases.

From the results obtained it is clear that a system should be designed based upon the type of traffic the system has the carry. However, it is not possible to predict all the possible traffic scenarios while building the system so the system should adapt as per the traffic requirements. IS-136 system has a fixed frame structure which is the reason why a change in the airlink frame loss probability, or a change in traffic mix effects the system performance. In the new system, the system adapts the frame structure with a change in the airlink frame loss probability and it controls the number and type of traffic requests in the system, hence, the system is immune to the change in traffic conditions or a change in airlink frame loss probability.

6.2 Adaptive Call Admission Control Algorithm

The reliability of the system in meeting the QoS requirements promised to the users depends upon the number and the type of call requests in the system. It is not possible to characterize all the traffic requests that the system can expect. The best that can be done is map a traffic request to a similar known traffic pattern and based upon the observed traffic characteristics the traffic estimates are updated to have a realistic or a close estimate of the traffic characteristics. In this dissertation we proposed a Call Admission Control (CAC) algorithm which uses the stored statistical arrival rate, service rate and the queuing delay of each class of service request to estimate the expected queuing delay. Upon arrival of a new call request the queuing delay is estimated by using a quadratic fit of the delays obtained from the statistical values and the delay estimate when a M/M/1 queuing model is used. In the CAC algorithm to compute the queuing delay and the load on the system the peak rate of

the new traffic request is considered. This can cause the system to be underutilized if the traffic is not smooth (peak and mean data rate of the traffic request are close). We compensate for the overestimate of the delay by dropping a call request with probability P_{drop} even when the queuing delay for different classes of service will be more than the allowed delay if the call is admitted. Given performance of wireless links $w.r.t$ the frame loss probability which is worse as compared to wireline links the system has to adapt the estimated service rate which a new call request is going to see to compensate for the link characteristics.

We studied the performance of the new call admission control algorithm for a traffic mix of two or three classes, different call arrival rates, and for different P_{drop} values. The performance of the CAC algorithm is evaluated by computing the ratio of calls dropped and for each class of service the ratio of the number of frames during which there was a delay violation to the total number of frames during the simulation period. For P_{drop} values close to one, with an increase in airlink frame loss probability the number of calls dropped increases but there is no change in the ratio of call violations for different class requests. For small values of P_{drop}, with an increase in airlink frame loss rate there is no change in the number of calls dropped, but there is an increase in the ratio of delay violations for different classes of traffic.

For an increase in call arrival rate the number of calls dropped is bounded by P_{drop} and the ratio of delay violations for different classes of traffic is different for different P_{drop} values. This is because the classes of traffic are classified as per the delay they can tolerate. Hence, lower class traffic which can tolerate higher delay values do not suffer more call violations as compared to higher class traffic requests. However, while scheduling the packets, packets belonging to a class of traffic with low interpacket delay are given priority over packets of traffic requests with higher

interpacket delay. Hence, lower class traffic are effected the most $w.r.t$ the delay violations. This also implies that, if it is possible to control the number of higher priority class traffic from being admitted to the system not only the delay violations of that class of traffic but the delay violations of lower priority class traffic can be controlled. This can be achieved by having different P_{drop} for different classes of traffic. For a traffic mix of four classes of traffic we run experiments for fixed value of P_{drop} and compare the performance with the algorithm where we have different value of P_{drop} for different classes of traffic, we call this as adaptive P_{drop} algorithm.

Adaptive P_{drop} algorithm adjusts the P_{drop} for different classes of traffic and ensures that lower priority classes traffic requests do not have a higher delay violation ratio due to an increase in the delay violation of higher priority class call requests or an increase in the number of higher priority call requests in the system. Moreover, this formulation is for any type of traffic request as we did not present the results for any predefined traffic properties of the call requests.

6.3 Future Work

In this thesis we addressed the problem of how to utilize the scarce wireless bandwidth efficiently. We provided a framework which ensures the system to provide the promised QoS to the users and also control the number of users admitted to the system. This work needs to be extended taking the users mobility into consideration. Different users have different mobility patterns and different applications have different QoS requirements. Based upon the user requriements proper the bandwidth has the be managed efficiently to handle call requests originating in the cell and call requests of users moving into the cell from an adjacent cell.

BIBLIOGRAPHY

[1] V. K. Garg, J. E. Wilkes, "Wireless and Personal Communications Systems", *Prentice-Hall Publications*, 1996.

[2] R. Seifert, "Gigabit Ethernet Technology and Applications for High-Speed LANS", *Addison-Wesley*, 1998.

[3] S. Keshav, "An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network", *Addison-Wesley*, 1997.

[4] E. Nikula, A. Toskala, E. Dahlman, L. Girard, A. Klein, "FRAMES Multiple Access for UMTS and IMT-2000", *IEEE Personal Communications*, April 1998, pp. 16-24.

[5] M. S. Taqqu, V. Teverovsky, W. Willinger, "Estimators for Long-range Dependence: An Emperical Study", *Fractals*, Vol. 3, No. 4, 1995, pp. 785-788.

[6] S. Sahu, V. Firoiu, D. Towsley, J. Kurose, "Traffic Models and Admission Control for Varible Bit Rate Continuous Media Transmission with Deterministic Service", *Proceedings of Performance and Control of Network Systems II, SPIE* 1998.

[7] S. Vamvakos, V. Anantharam, "On the Departure Process of a Leaky Bucket System with Long-Range Dependent Input Traffic", Queueing Systems: Theory and Applications, Vol. 28, Nos 1-3, pp. 191-214, May 1998.

[8] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.,* vol. 8, pp. 368-379, Apr. 1990.

[9] D. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," in *Proc. IEEE Tricom '91,* pp. 35-43 1991.

[10] H. Zhang and D. Ferrari, "Rate-controlled static-priority queueing," in *Proc. IEEE INFOCOM '93,* 1993 pp. 227-236.

[11] L. Zhang, "VirtualClock: A traffic control algorithm for packet switching networks," in *ACM Trans. Comput. Syst.,* vol. 9, pp. 101-124, May 1991.

[12] A. K. Parekh and G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Networking, vol. 1,* pp. 344-357, June 1993.

[13] S. J. Golestani, "A Stop-and-Go queuing framework for congestion management," in *Proc. ACM SIGCOMM '90,* pp. 8-18 September 1990.

[14] S. J. Golestani, "Duration-limited statistical multiplexing of delay-sensitive traffic in packet networks,", *Proc. IEEE INFOCOM '91,* pp. 323-332 1991.

[15] C. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks," in *Proc. IEEE GLOBECOM '90,* pp. 300.3.1-300.3.9. 1990.

[16] N. R. Figueira, "Leave-in-time: A new service discipline for control of real-time communications in a packet-switching network," in *Proc. ACM SIGCOMM '95,* pp. 207-218 Aug-Sep 1995.

[17] H. Zhang and D. Ferrari, "Improving Utilization for deterministic Service in Multimedia Communication", *IEEE International Conference on Multimedia Computing and Systems,* 1994.

[18] H. Zhang and E. W. Knightly, "Providing End-to-End Statistical Performance Guarantee with Bounding Interval Dependent Stochastic Models", *Proc of ACM SIGMETRICS '94,* pp. 211-220, May 1994.

[19] W. Verbiest, L. Pinnoo, and B. Voeten, "The Impact of the ATM Concept on Video Coding", *IEEE Journal of Selected Areas in Communication,* 6(9):1623-1632, Dec. 1988.

[20] J. Y. Hui, "Resource Allocation for Broadband Networks", *IEEE Journal of Selected Areas in Communication,* 6(9):1598-1608, Dec. 1988.

[21] H. Saito and K. Shimoto, "Dynamic Call Admission Control in ATM Networks", *IEEE Journal of Selected Areas in Communication,* 9(7):982-989, Sept 1991.

[22] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in high Speed Networks", *IEEE Journal of Selected Areas in Communication,* 9(7):968-981, Sept. 1991.

[23] F. P. Kelly, "Effective Bandwidths at Multi-Class Queues", *Queuing Systems,* 9:5-16, 1991.

[24] D. D. Clark, S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", *Proc. of ACM SIGCOMM '92,* Aug. 1992.

[25] S. Jamin, S. Shenker, L. Zhang, and D. D. Clark, "An Admission Control Algorithm for Predictive Real-Time Service (Extended Abstract)", *Proc. 3rd Int'l Network and Operating Systems Support for Digital Audio and Video Workshop,* Nov. 1992.

[26] S. Jamin, P. B. Danzig, S. Shenker, L. Zhang, "A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks", *Proc. of ACM SIGCOMM '95,* pp. 2-13 Aug-Sep 1995.

[27] R. L. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation", *IEEE Transactions on Information Theory,* 37(1):114-131, Jan. 1991.

[28] A. K. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks", Ph.D thesis, MIT, Lab, for Information and Decision Systems, Tech. Report LIDS-TR-2089 1992. parts of this thesis were also published in the *ACM/IEEE Transactions on Networking,* 1(3):344-357 and 2(2):137-150.

[29] X. Qui and V. O. K. Li, "Dynamic Reservation Multiple Access (DRMA): A New Multiple Access Scheme for Personal Communication System (PCS), *ACM Journal on Wireless Networks*, Volume 2, No. 2 (1996), pp. 117–128.

[30] P. Roorda and V. Leung, "Dynamic Time Slot Assignment in Reservation Protocols for Multiaccess Channels", *proceedings of IEEE Pacific Conference on Communications, Computers and Signal Processing* (1993), pp. 451-454.

[31] J.-P. M.G. Linartz, "Packet-Switched Cellular Communication Architecture for IVHS using a Single Radio Channel", *Proceedings of IEEE PIMRC* (1994), pp. 1222–1226.

[32] T. Cheng and H. Tawfik, "Performance Evaluation of Two Channel Assignment Algorithms in Cellular Digital Packet Data Networks," *Proceedings of IEEE PIMRC* (1995), pp. 537–543.

[33] D. A. Wismer and R. Chattergy, *Introduction to Nonlinear Optimization: A Problem Solving Approach*, North-Holland.

[34] A. I. Elwalid and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks", *IEEE Transactions on Networking* Vol 1, no 3, June 1993, pp. 329–343.

[35] W. Lau and S. Li, "Statistical Multiplexing and Buffer Sharing in Multimedia High-Speed Networks: A Frequency-Domain Perspective", *IEEE Transactions on Networking* Vol 5, no 3, June 1997, pp. 382–396.

[36] G. Wu and J. W. Mark, "Computational Methods for Performing Evaluation of a Statistical Multiplexer Supporting Bursty Traffic", *IEEE Transactions on Networking* Vol 4, no 3, June 1996, pp. 386–397.

[37] M. H. Chan and J. P. Princen, "Prioritized Statistical Multiplexing of PCM Sources", *IEEE Transactions on Networking* Vol 3, no 5, October 1995, pp. 549–559.

[38] D. D. Clark, S. Shenker, L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", Proceedings of *ACM SIGCOMM '92*, August 1992, pp. 14–26.

[39] *Digital Cellular Telecommunications System (Phase 2+); General Packet Radio Service (GPRS); Service Description; Stage 2; (GSM 03.60).* GSM Technical Specification, February 1997.

[40] *Digital Cellular Telecommunications System (Phase 2+); General Packet Radio Service (GPRS); Mobile Station (MS)-Base Station System (BSS) Interface; Radio Link Control/Medium Access Control (RLC/MAC) Protocol (GSM 04.60 version 6.0.0).* Draft EN(04.60), March 1998.

[41] Naveen K. Kakani, S. K. Das, S. K. Sen and M. Kaippallimalil, "Optimizing QoS-based Channel Allocation in Wireless Data Packet Networks," *Proceedings of the 7th IEEE Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks* (CAMAD'98), Sao Paolo, Brazil, pp. 80-88, Aug 1998.

[42] Naveen K. Kakani, S. K. Das, S.. K. Sen, and M. Kaippallimalil, "A framework for Call Admission Control in Next Generation Wireless Networks," *Proceedings of the First ACM International Workshop on Wireless Mobile Multimedia*, Dallas, pp. 101- 110, Oct 1998.

[43] M. Naghshineh, M. Schwartz, A.S. Acampora, "Issues in Wireless Access Broadband Networks", *Wireless Information Networks*, edited by J.M. Holtzman, Kluwer Academic Publishers, 1996.

[44] D. Newman, K. Tolly, "Wireless LANs: How Far? How Fast?", *Data Communications on the Web*, http://www.data.com/Lab_Tests/Wireless_LANs.html (23rd September 1999).

[45] "UCB/LBNL/VINT Network Simulator - ns (version 2)", http://www-mash.cs.berkeley.edu/ns/ (10th October 1999).

[46] "Linear Programming Software lp_solve2.0", http://mat.gsia.cmu.edu/GROUP95B/0231.html (20th October 1999).

[47] G. Brasche and B. Walke, "Analysis of Multi-Slot MAC Protocols Proposed for the GSM Phase 2+ General Packet Radio Service", *IEEE VTC*, pp. 1295-1300, 1997.

[48] P. Taaghol, R. Tafazolli, and B. G. Evans, "An Air Interface Solution for Multi-rate General Packet Radio Service for GSM/DCS", *IEEE VTC*, pp. 1263-1267, 1997.

[49] J. Cai, and D. J. Goodman, "General Packet Radio Service in GSM" *IEEE Communications Magazine*, pp. 122-131, October 1997.

[50] G. Brasche, "Evaluation of a MAC Protocol Proposed for a General Packet Radio Service in GSM", *IEEE VTC*, pp. 668-672, 1996.

[51] R. Ludwig and D. Turina, "Link Layer Analysis of the General Packet Radio Service for GSM", *IEEE VTC*, pp. 525-530, 1997.

[52] R. Nelson, *Probability, Stochastic Processes and Queuing Theory*, Springer-Verlag, 1996

[53] E. Nikula, A. Toskala, E. Dahlman, L. Girard, A. Klein, "FRAMES Multiple Access for UMTS and IMT 2000", *IEEE Personal Communications*, pp. 16-24, April 1998.

[54] T. Ojanpera et al., "Comparison of Multiple Access Schemes for UMTS", *Proceedings of IEEE VTC*, Phoenix, AZ, May 1997.

[55] Uyless Black, "Mobile and Wireless Networks", Prentice Hall Series in Advanced Communications Technologies, 1996.

[56] Uyless D. Black, "Advanced Internet Technologies", Prentice Hall Series in Advanced Communications Technologies, 1999.